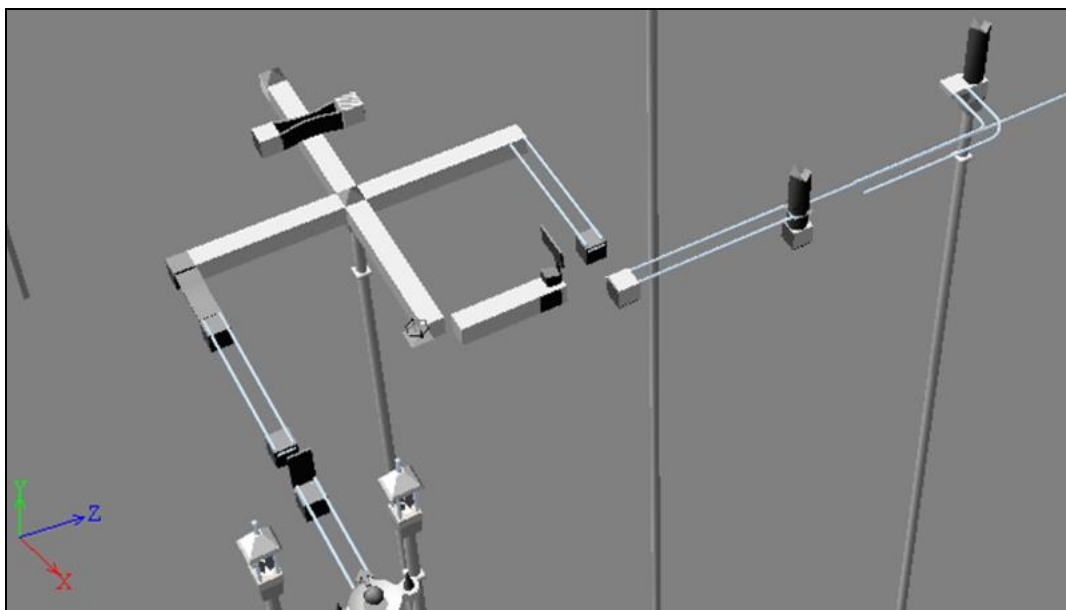


Virtools 制图

GEcoldstar, BallanceBug 编著

GCBA 官方认证教材 Virtools 制图

——第二版——



GCBA 《Virtools 制图》 第二版 第 1 次修订

修订时间：2019 年 2 月 11 日

字数：23 千字

网址：<http://gcba.ml/>

前言

欢迎阅读此教程。在阅读此教程之前，你首先要简单了解 Ballance 游戏本身的原理和制图的简要流程。

Ballance 是 Virtools 开发的游戏，可以直接生成和编辑地图文件（Ballance 的地图扩展名是 NMO），但鉴于功能性等多方面原因，Virtools 常用于后期对地图进行修改，而建模往往要利用 3ds Max 等建模功能强大的软件，少有完全只用 Virtools 制作地图的作者。

但 Virtools 在 Ballance 制图中是最为重要的，可以这么说，只用 Virtools 也可以做出一张地图，但如果离开了 Virtools，几乎不可能做出一张地图，可见这部分的重要性和基础性。当然，对 Virtools 制图的原理的理解也是制图最需要具备的素养。

本教程旨在综合运用 Virtools 修改和制作 Ballance 地图，并且提供少量可以修改游戏设置的方法。通过认真理解本书对于 Virtools 制图的整体观和具体操作，读者可以领略 Virtools 制图，并且有一定制作 Ballance 地图的能力。

欢迎经验丰富的人不断完善该教程，我们致力于打造最全面、最易于接受的地图制作教程。

让我们开启 Ballance 制图之旅吧！祝你在 Virtools 中打开新世界的大门！

目录

第一章 了解 VT	4
1.1 VT 的注册与初始界面	4
1.2 地图的导入与导出.....	6
1.3 一些基本操作.....	7
第二章 物件与组	12
2.1 物件概述	12
2.2 物件归组	15
2.3 小节组改换	18
第三章 VT 元素概述	20
3.1 网格概述	20
3.2 材质与贴图	21
3.3 素材的导入与导出.....	23
3.4 光源简介	25
第四章 VT 元素拓展	28
4.1 用什么表达 3D.....	28
4.2 路面加影子	30
4.3 贴图的内置和外置问题	32
4.4 RSC 目录的导入和应用	34
第五章 游戏机制与制图	36
5.1 对机关的总结.....	36
5.2 制作好一张地图	37
5.3 3D Entities 目录中的配置文件.....	40
后记	43

第一章 了解 VT

要想了解 Ballance 制图，就必须从了解基础的工具入手。对 Virtools 的初步了解是学习 Virtools 制图的第一步，我们从这里开始了解 Virtools（之后简称 VT）。

1.1 VT 的注册与初始界面

各种版本的 Virtools 直属文件夹里面都有个“Dev.exe”（其实真正的 VT 程序是 devr.exe，Dev.exe 只不过是到真正 VT 程序的链接程序，是习惯上的打开方式。如果兼容性有误，需要设置 devr.exe 的属性，兼容模式设成 Windows XP (Service Pack 3) 即可），打开后需要去注册。

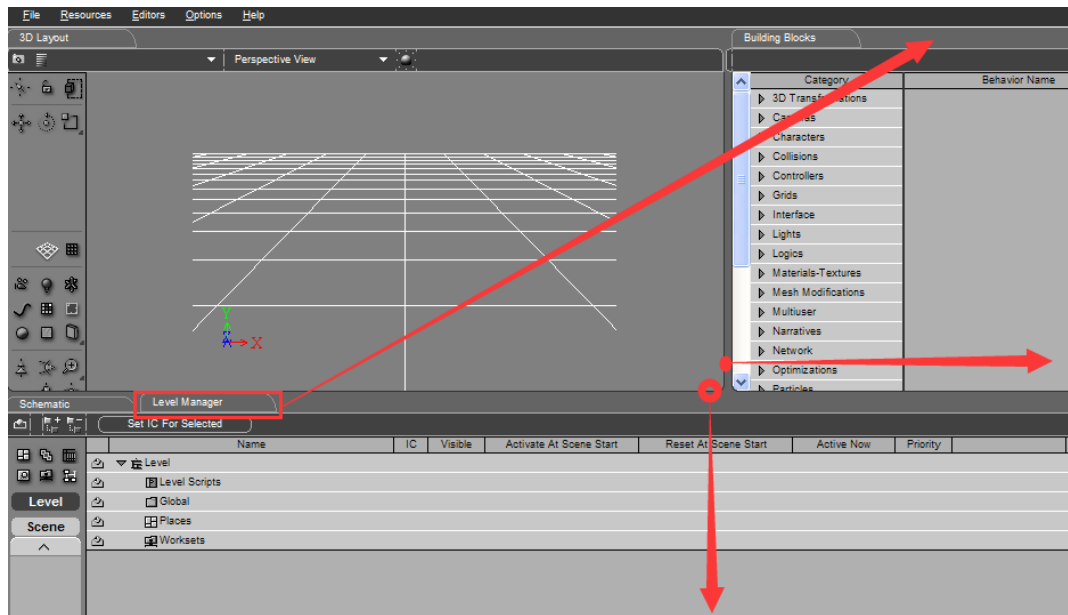
注册时它会给你两个选项，选择第二项“specify the licence file”，意为寻找注册文件，之后弹出的框中，点击方框下的“Browse”，浏览目录，找到你通过各种途径下载的文件“license.dat”（或 license.lic，授权文件的名称也可能是 licence 等其他名称），并且选择，确认，即注册完成。（注意：注册时的第一项需要输入注册码，目前尚未找到可用注册码，因此不可行）

另外因为不同版本 VT 对授权文件的信息要求不同，所以换用不同版本的 Virtools 时，需要手动更改下授权文件对应版本，否则就会显示授权文件有误，而无法打开其他版本的 VT。不过如果用了三版本（3.0、3.5、5.0）通用的授权文件，就不用考虑这个了。

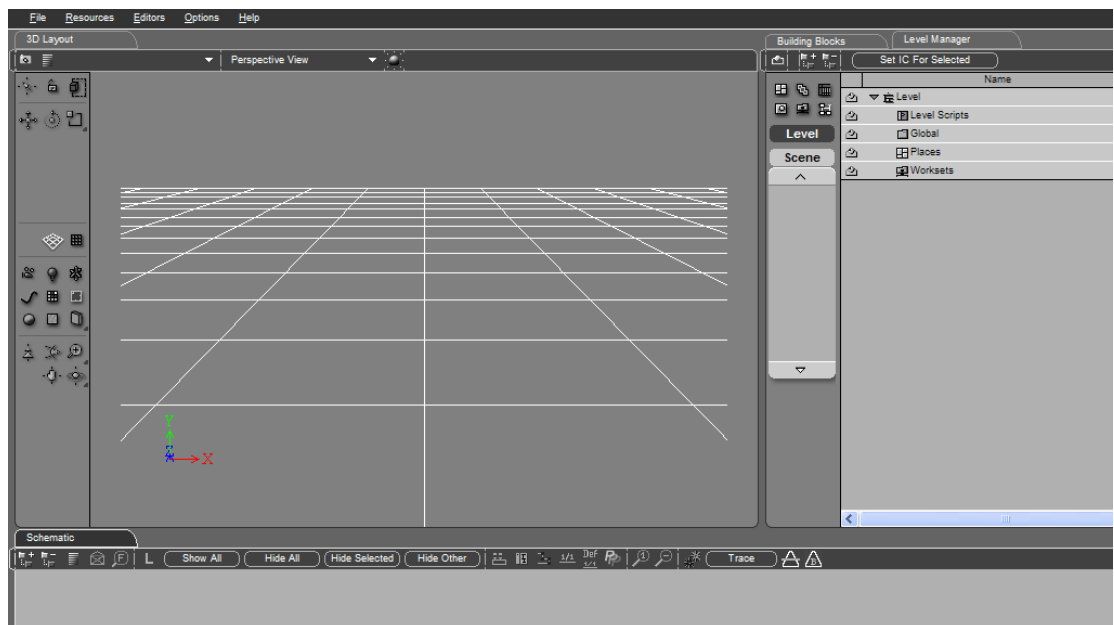
Virtools 3.0 是现有最适合制图的版本，但多数电脑尤其是 win8/10 注册 3.0 版本很容易抽风，因此用 3.5 版本制图更加广泛，不过 3.5 版本也存在一些电脑打不开的问题。Virtools 5.0 做出来的图无法被 Ballance 识别，加载时会卡进度条，但是用 3.0 或 3.5 版本的 Virtools 打开并保存一次即可正常加载。5.0 版本虽然“不适合制图”，但在一些其他方面有用（如能实现导出 obj 而与 3ds Max 交流的插件 vt2obj，这里不再介绍），而且兼容性最好。Virtools 5.0 打不开的情况可以尝试通过安装一个适配自己电脑的 DirectX 修复器解决。

注册完成后，要想修改 Ballance 地图，需要先了解 Virtools 的界面。首先说明一点，目前暂未能找到汉化版的 VT，但是其实基本不需要英文能力——VT 的英文并不是非常难懂，而且很多是缩写，英语好的人也不一定能理解。何况适用于制图的操作熟悉的是方法，记住怎么用对应功能即可，因此英文版的 VT 习惯了也就适应了。

接下来让我们看看初次打开 VT 后的界面效果以及建议做的第一步操作：



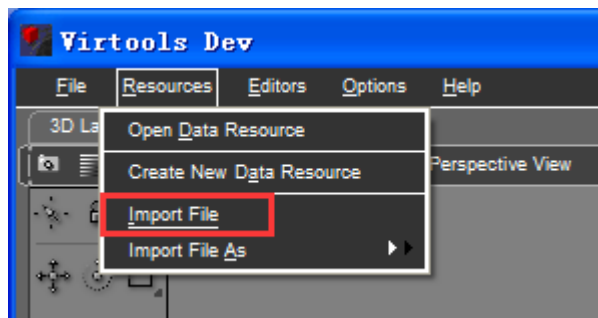
我们会看到这样的景象——屏幕被分成了三栏，左上默认是 3D Layout 选项卡（以后简称 3D 框），主要用于效果演示与输出；右上默认是 Building Blocks 选项卡（简称 BB 列表），用于查找脚本，对于制图用处不大。下侧默认是 Level Manager 选项卡，是主要检索关卡元素的一栏，非常重要。3D 框看起来有点小了，可以先拉大一点，按住边界拖动就行了。Level Manager 选项卡里的元素在制图时会显示很多，又因为右边初始的 BB 列表对于制图用处不大，所以可以把 Level Manager 框拖动到右上一栏（按住“Level Manager 选项卡”拖动即可），这样后续处理会方便得多，另外，双击选项卡会切换该选项卡为全窗口显示，再双击又会回归之前调好的界面。这样操作后，我们看到的便是这样了：



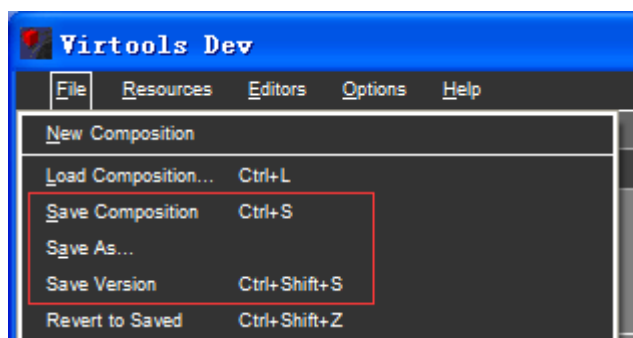
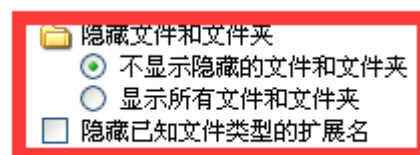
不过每次打开 VT 后都可能得重新进行拖动 Level Manager 的操作，习惯即可，这样对制图相对比较方便。

1.2 地图的导入与导出

我们知道，Ballance 的地图文件的格式是 nmo，但是 VT 中却只能打开 cmo 文件。别急，其实 nmo 文件也是 VT 可识别的文件，只不过是以 VT 素材的形式而不是 VT 关卡的形式出现的。VT 关卡的默认格式是 cmo，而最支持的素材的默认格式是 nmo。所以我们要以导入素材的形式打开 nmo 文件，具体操作如右图。



Ballance 地图其实是伪装成 nmo 格式的 cmo 文件（进一步研究表明，由于 cmo 不仅支持地图，还支持动画，而 nmo 是场景中静态对象的最好载体，所以地图对外用的是 nmo 格式，不过改成 cmo 来编辑确实能带来不少便利），所以可以把要编辑的自制地图扩展名直接改



成 cmo，用 VT 编辑完后直接以 cmo 的形式保存，要玩的时候再改成 nmo。如果你的电脑把 cmo 设成 VT 默认编辑文件并且隐藏扩展名，试试设置文件夹选项（这个不同系统各异，在这里不作介绍），显示已知文件类型的扩展名即可修改扩展名。

地图直接以 cmo 形式保存有三种保存形式：Save Composition、Save As... 和 Save Version。Save Composition 相当于通常我们编辑文档用的“保存”，Save As... 相当于通常我们编辑文档用的“另存为...”，而 Save Version 比较特殊，直译为保存版本，Save Version 可以保存修改后的地图，同时不替换修改前的地图。

举个例子，当打开一个名为“2HAD.cmo”的地图文件并编辑修改后，用 Save Version 可以新保存一个名为“2HAD_01.cmo”的地图文件，而原先没被改动的文件依然能被保存，同理这样还可以保存“2HAD_02.cmo”、“2HAD_03.cmo”等文件，

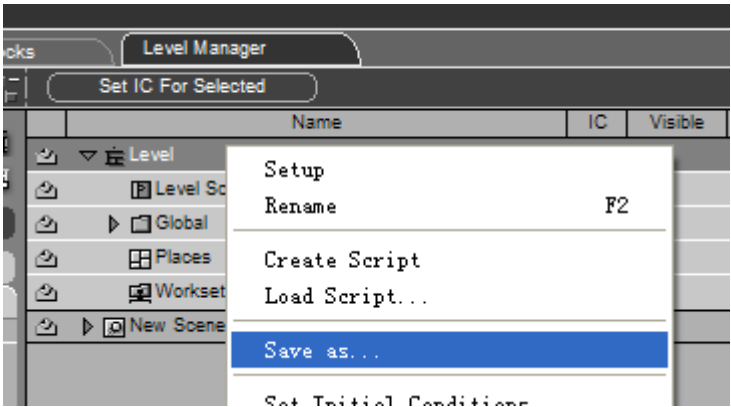


如图所示。

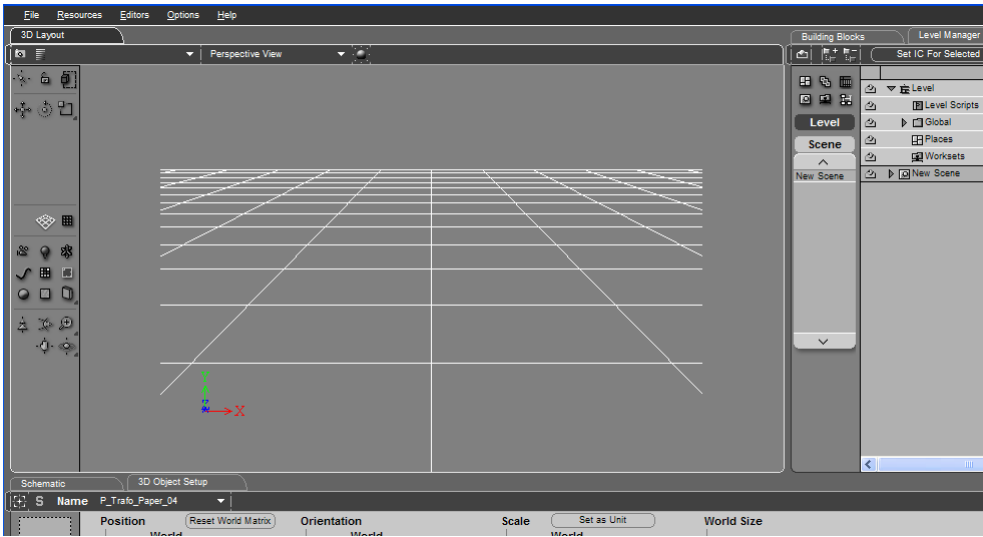
注意：原版地图与绝大多数自制地图不一样，不能通过改成 cmo 直接打开的方式打开，否则会出错。请用导入 nmo 文件的方法导入，保存以后就可以随意用 cmo 文件的编辑方法了。

地图也可以直接导出成 nmo 格式，只需在 Level Manager 中的根目录 Level 上点右键“Save As...”即可，如图所示：

我们还会遇到一类 nmo 文件。这些文件不是关卡，也不是游戏里面特殊的配置文件，这些文件被称为素材。素材的导入应该使用本节第一段的方法。关于素材的导入和导出有较深的学问，此处不再讨论，详见本书第三章第 3 节。



1.3 一些基本操作




我们将沿用本章第 1 节调整过后的界面对本节内容进行介绍。有一个不方便的是，每次打开 VT 都需要你去把界面调整成那样，但是这些无关紧要，调了还是为了编辑地图的方便。让我们再一次观察一下导入地图之后的 VT 界面：






为什么什么都没有？其实这不是你的错。每次打开 VT 后，3D 框中都会把你的视角位置调成虚拟世界的最中心，但是不是所有地图都有在最中心的物体。要想找到地图物体，需要你进行换视角位置等的一系列操作。3D 框左边的一系列图标相当于工具栏，左图为各分区图标的大体作用。


如下是对这些小图标中常用图标的详细介绍：






：选此图标后，鼠标回到 3D 框内，按住鼠标往下或上挪，镜头就会扩张或缩小，一般遇见未知地




图可用此图标放大视野来找到地图。


和：隐藏或显示 VT 界面里默认给的栅格，觉得默认栅格碍事的改图者可以用这两个图标隐藏。




：选此图标后，鼠标回到 3D 区域，可以移动镜头。


：以环绕的方式调整镜头的工具。当你未选中任何东西时，用这个工具旋转相当于围绕坐标中心旋转，这种不选中物体的旋转一般不常用，因为镜头会乱跑。但是如果选中了物体后再用这个工具，拖动鼠标，镜头便会绕着这个物体旋转。

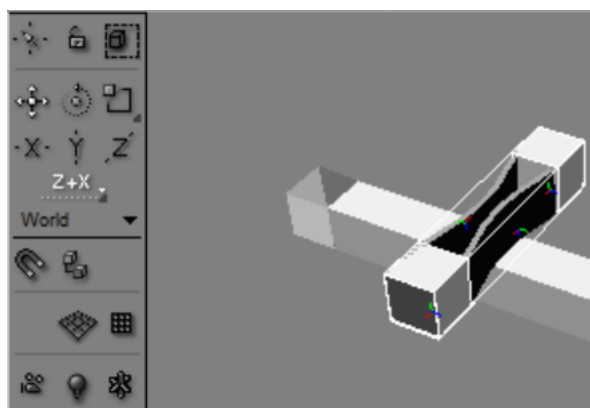
可以试试在上按住鼠标，会出来几个选项：，拖动鼠标至上，松开，就变成了。这时使用这个工具，将直接旋转摄像机的方向。




：用于视角前后移动，与功能相似。但是如果按住鼠标改为后，选中物体后点击，则会将视角自动移动至刚好能够容纳该物体的位置。此功能比较有用，有时可以快速定位物件/在 3D 框场景中瞬移。

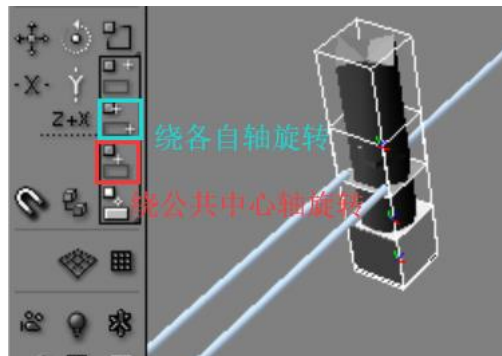
：选定物体；鼠标拖动可选定涉及区域内的批量物体；按住 Ctrl 键配合这个使用可以选择性的选定多个物体，此时需要了解一件事：

指的是只要选择框选到了该物体的一部分即可选中该物体。此操作极易引起误选，因此建议点击一下改为。此时，只有物体完全处于选择框内，该物体才会被选中。不过还是建议在选中后点击查看一下是否存在误选。


：移动物体。在选中物体后，可以选择移动物体的基准坐标轴或两个坐标轴形成的平面。例如下图所示的物体的移动，选择 Z+X 平面后物体只能在 Z+X 平面移动，而物体不会在 Y 轴方向移动。





：旋转物体，和移动物体相似，可以选择基准坐标轴或平面进行旋转，不过旋转物体选择的基准轴或平面较难确定。我们可以点击，使其变成，这样就可以锁定单次旋转角度，旋转时幅度只能旋转设定角度的整数倍（设定角度默认是 10° ，可以自己在该按钮上右键调整为任何想要的角度 [参数名为 **Rotate Step**]，建议调成 22.5° ），这样可以准确的旋转物体。此外，旋转选定的多个物体时还可以调整旋转方式，可以选择物体绕各自轴旋转，也可以选择绕公共轴旋转。



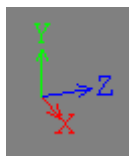
事实上我们在本书的下一章就会学到如何通过具体调整物件参数来移动或旋转物体。

：扩大或缩小物体。此工具在 **Ballance** 地图修改或编辑过程中一般不常用，且容易出现一些问题。

需要强调的是，对物体使用这些操作时必须鼠标放在物体上，否则容易出现误选物体等操作问题，此时可以通过点击把需要操作的物体锁住，此时点击屏幕中任意位置均可对该选中物体操作。

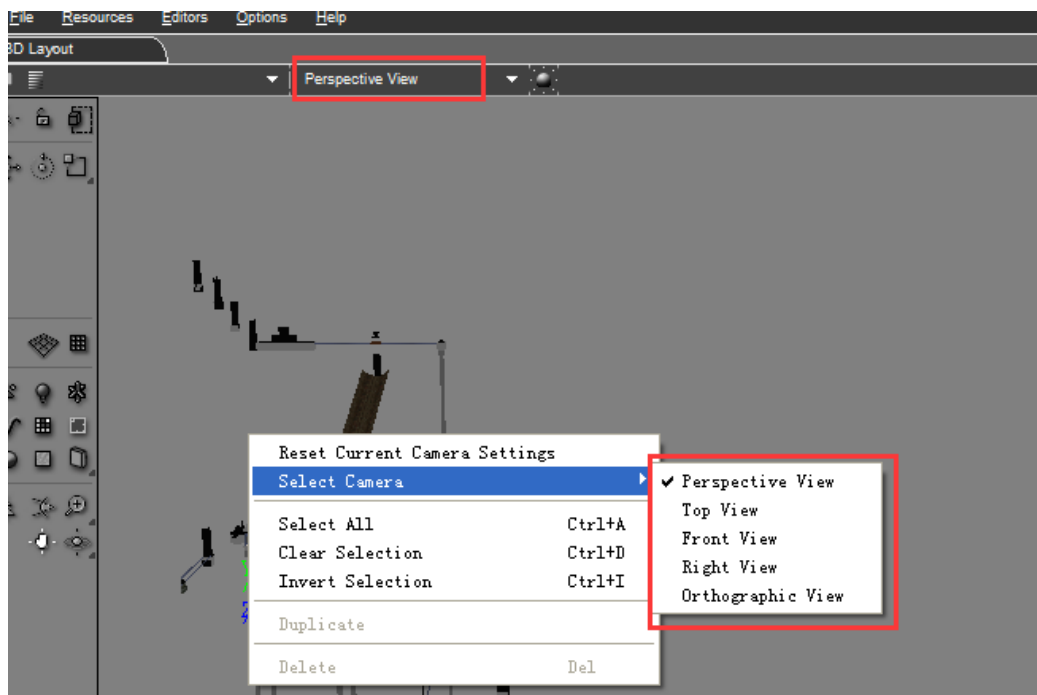
：在镜头视野前方放置一个灯泡。利用此工具可以给地图进行照明，但是灯泡毕竟不是游戏中固有的元素，而且含有灯泡的地图可能会出现其他不太好的影响，所以在照明完后应当删除灯泡（严格地说应该把灯泡称作点光源，这只是光源的一种，我们将在本书第三章第 4 节详细了解光源）。

至于其他的工具，对编辑和修改 **Ballance** 地图帮助效果不大或过于复杂，此处不再介绍，有兴趣的可以自行探索。



为坐标轴，代表地图空间中的三个方向，这是定位物体的基础。接下来的镜头变换内容中我们可以看到坐标轴的展示发生了与镜头变换相应的一些变化。



另外，右键点击 3D 框后弹出的菜单中可以选择不同类型的镜头，如图所示：

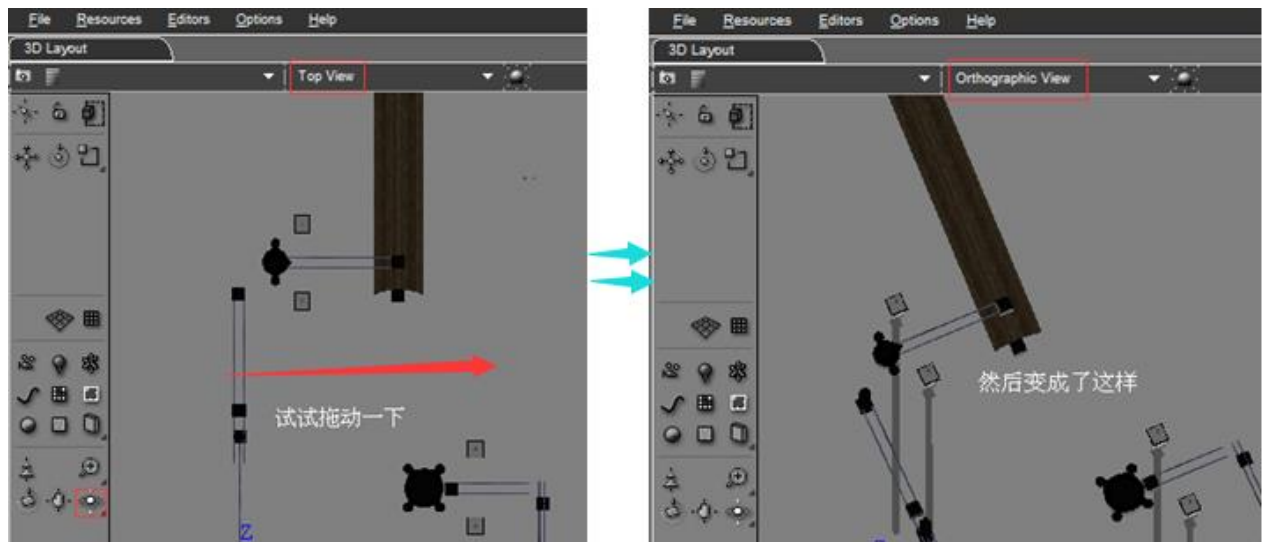


默认镜头是 **Perspective View**，译为透视视图。透视视图存在消失点，能体现远小近大的规律，是我们美术上用的视图形式，也是我们身临其境时眼见的真实视图形式。这是我们最常用的镜头。我们可以发现镜头的状态在图上标出的红色框中有显示。我们可以通过右键换其他镜头来实现对地图的精准编辑。


Top View 是“俯视图”的意思，即从上往下看，选用该镜头可以准确调整物体在 $X+Z$ 平面的位置，但无法调整 Y 轴位置（即竖直方向的位置）

同理，**Front View** 和 **Right View** 分别表示正视图和右视图。利用它们，我们也可以较为准确的对准物体相对于与坐标轴平行或垂直的平面的位置。

Orthographic View 是正交视图的意思，当 **Top/Front/Right** 视图经过  等能改变视图中坐标轴方向的变换后，3D 框的镜头就会立刻转变为正交视图。正交视图中所有网格线平行，是我们数学上用的视图形式，因此上述操作有利于观测和检验用标准视图对准后的物体的位置。作图时惯用这个镜头可能有些违和感，尤其是柱子很多的地图。当再用 **Top/Front/Right** 视图的时候再次用  变换镜头，**Orthographic View** 的镜头位置就会瞬间改变。当刚进入地图且没有上述所说的镜头变换时，**Orthographic View** 展示是效果与 **Top View** 保持一致。



另外，我们还可以利用一些快捷键来实现一些便捷操作：

滚动鼠标中轴：放大或缩小视野（放大视野有一定的限度，可以先缩小然后用  放大，就可以提高滚动鼠标中轴放大视野的限度）；

在 VT 界面中点击键盘上的 Y 键，视角可以根据鼠标在同一高度内自由转动。如果想提高/降低视角，点击鼠标右键拖动；如果想转的快一些或者到其他高度，使用鼠标左键。不过需要注意的是，视角移动速度取决于按 Y 键时鼠标在画面内的位置。鼠标越远离初始位置，视角移动速度就越快。

按住鼠标中轴并移动鼠标：移动镜头；

按 alt 再按住鼠标中轴并移动鼠标：绕某个中心旋转镜头，（刚打开一个界面时旋转中心是坐标中心）。

ctrl+z：撤销，但只能撤销在 3D 框中进行的操作，之后学到的 Level Manager 操作中就用不了撤销操作了，所以需要谨慎。

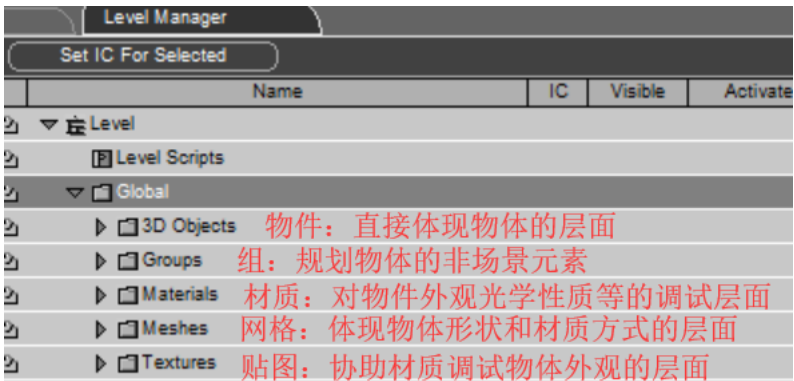
第二章 物件与组

当我们熟悉了工具栏中的操作工具和镜头的属性后，难免要问，我们移动的物体是什么？确实，在上一章我们只讨论了在场景中操作的简要方法，但是不涉及物体本身的性质和原理。我们将在本章讨论这些问题。同时，我们也会在本章渗透关于组的概念以及应用。

本章内容可能略难懂，但却是有原因的。毕竟学习制图的你可能还不理解物体这个整体概念，也不了解其包含的层次中有各种从属的原理——物件、网格、材质、贴图，甚至对组的概念也模糊不清。放轻松，保持认真自信，在理解本章内容和继续了解之后的内容后，形成对 VT 的整体观念，就会明白这些的。

2.1 物件概述

首先，我们要看一看 Level Manager 中的元素，点开里面的 Global，我们可以看到一系列东西，列表中就是我们的元素的“根”，以有规范的名称和参数的形式在 Level Manager 里储存，而在 3D 框内看到的只是元素在场景的表现形式。



如图所示，在 VT 中组成地图的元素可分类为物件、网格、材质、贴图和组。其中物件（3D Object）是物体体现的最直接形式，也可以直接和 3D 框中的物体对应，而本节我们就讨论互相认定表格和场景中的物件（本章暂不讨论网格、材质和贴图，只需要知道它们与物体本身关联紧密）。

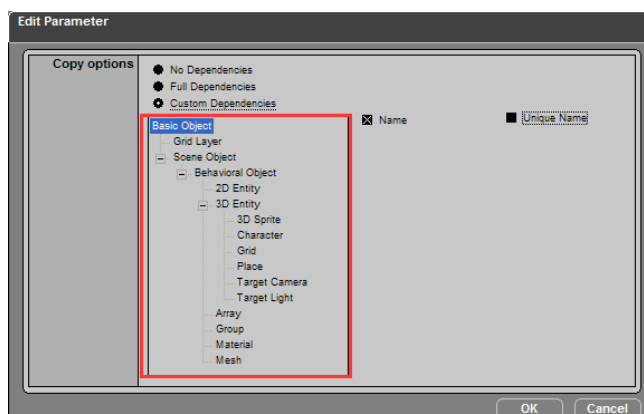
点开 3D Objects 物件列表目录，会展示出地图中的所有物件。每个物件后面都有一个像眼睛一样的东西，有 、、 三种模式，第一种为可见，第二种为隐藏，第三种为隐藏本身及子对象；第一种是大部分物体的状态，第二种是云层（或漩涡）和死亡区，以及某些地图中用来“坑人”的隐形墙的状态，第三种基本不用。*[不过请注意：机关类物件（由游戏本身脚本机制渲染而可以动的物件）即使在 VT 里设为隐藏，在实际游戏里仍会显形。只有非机关类物件设置隐藏才有用。（关于机关类物件，有更深层次的问题分析，此处仅作了解，详情请见本章第 2 节以及后续章节对其的重新定义和解释）]*

在之后，为了方便描述，我们将把在 3D 框中可对应 Level Manager 中 3D Object 物件的物体统称为物件。

度会直接影响地图的合理性。可以利用调整合适的镜头手动对齐校准，也可以在一些特殊情况下通过调整物件坐标来对齐（受物件的轴的位置影响），还可以使用 VT 中的 Align 功能（形状对齐，不看物件轴的位置）。选中多个物体后，在 3D 框里右键 Actions - 3D Entities - Align 可以弹出对齐设置，可以选择对齐的轴和对齐方式：Front 是该轴正方向上多个物件平齐，Center 是中央对齐，Back 是该轴反方向上多个物件平齐。对齐操作会保持选择时的第一个被选中的物件位置不变，而其余的物件位置改变。

组(Groups)是一个和场景无关的元素，仅在 Level Manager 中存在，但组非常重要，因为这是物体各种属性的必要条件。在 Ballance 关卡中，除了少数装饰性的物件外，其余物件都要有所牵连的组。我们把将物件和组牵连的过程叫做“归组”。不归组的后果可能是球无视路面或钢轨而直接穿过，也可能是球在路面或钢轨上滚动或撞击无声，甚至可能打不开游戏（即加载关卡半途中卡死）。所以，归组具有非常重要的意义（我们将在本章第 2 节和第 3 节学到物件归组的知识）。当我们明白物件会有牵连组，以及未来将学到的网格、材质和贴图后，就可以学习如何复制或删除 VT 中的物件了。*[归组的机制会在之后介绍]*

我们先来讨论物件的复制。复制物件的方法有两种，一是在 3D 框中对选中的物件右键 Duplicate，然后会直接弹出一个选择框。二是在 Level Manager 中的物件列表中对物件右键 Copy 或快捷键 Ctrl+C，之后粘贴（快捷键 Ctrl+V）时就会弹出同样的选择框。这个选择框其实就是让你确定复制时应该同时附带复制的其他元素的。



由于物件、组、网格、材质和贴图是相互关联的，复制时需要选择关联和从属的复制，所以粘贴时才会显示这个框。如下是对应选择的意义：

No Dependencies: 没有从属——其关联对象均不选（即全不复制，但其实是用了 Custom Dependencies 里的预设），如果选了这一项，那么所复制的物件就没有从属复制组和网格，需要手动归组，归组参见本章第 2 节。

Full Dependencies: 全从属——其关联对象全选，选择这个选项的优点是什么都不用管，就可以正常使用了。但是全从属的缺点是可以共用并无相互影响的材质和贴图重复复制太多，可能会导致地图文件大小迅速增大。

Custom Dependencies: 自定义选择复制关联对象。如果复制的是左边红框中的某一类对象，就要注意右边与其对应的关联对象中有哪些要连带复制，建议选上的内容有：Basic Object 里的 Unique Name（单独命名），因为名字重复可以导致混淆，有时还会出现一些问题，这个选项选上后，重复名称会加上数字（如

001、002）以区分；Behavioral Object 里的 Groups Belonging (归组信息)，这样就自动归到原来的那个组了，不用再单独进行归组；复制路面、钢轨和死亡区时建议选上 3D Objects 里的 Meshes (网格)。[*详见本书第三章第 1 节。此处了解复制从属方法即可，有关需要理解的内容会在之后介绍]*

另外一定要注意的，VT 里选项上的“×”是对勾的意思，意思是选上了，千万不能被习惯性思维误导，以为是打叉。当选项为空时才是不选。

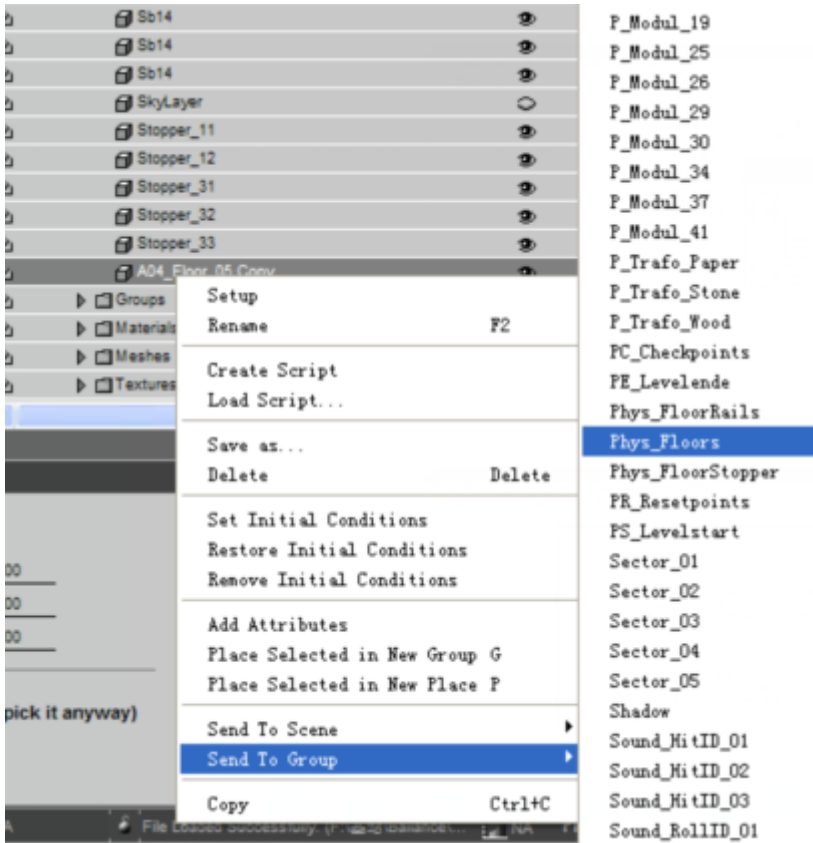
同样地，当我们删除 (Delete) 物件时，也会弹出这样的框，No Dependencies 是不删除任何关联对象，而 Full Dependencies 会连带删除物件所关联的全部网格、材质和贴图。*[因此，建议删除物件时选择 No Dependencies，等地图做完后再在 VT 上的 Options (选项)——Unused Objects Explorer (未使用对象浏览器) 中删除掉未被使用的对象 (对于制图者来说只会删除掉未被使用的材质、贴图和网格，而原地图中的灯泡和多余的空组需要制图者找到并手动删除)。]*

2.2 物件归组

当物件没有归到对应组时，或导入了素材物件 (素材物件不含归组) 时，就

需要对物件进行归组 (一个物件可以同时多个组里)。如图所示，我们要在 3D Objects 里找到需要归组的物件，点击右键 Send To Group，去找到相应的要归的组。(图中未给出全部的组，仅作演示)

组可以按具体情况分为路面组、机关组、小节组和声音组。顾名思义，路面组和机关组分别对应路面和机关，小节组规定了物件所在小节，而声音组规定了球撞击物件或在物件上滚动的声音。游戏通过对组名称的审查机制来决定组内物件的性质，因此归的组都是



是有固定名称的，如下是一般的归组规则：

路面应该归到路面组，并且同时要归到路面对应声音组（碰撞声音组和滚动声音组一般都要归）；机关（按上节的定义，分数球、道具球等动态道具也属于

机关)应该归到机关组,并且要归到对应小节的小节组(机关必须同时归机关组和小节组才能正常加载关卡!)但是声音组不能归给机关。*[游戏对归机关组的物件有特殊的性质改变机制,会渲染出机关对应的效果,且只能在当前小节激活。机关的原理在之后还会继续介绍。]*

下面来说一下各个组的信息,方便在归组时选取。

不动物件组: 路面组 Phys_Floors 钢轨组 Phys_FloorRails

死亡区组 DepthTestCubes

支撑机关稳定而对玩家球无效的物件归组 Phys_Floorstopper

机关组:

机关	对应组名	机关	对应组名
石球转换器	P_Trafo_Stone	木球转换器	P_Trafo_Wood
纸球转换器	P_Trafo_Paper	道具纸球	P_Ball_Paper
道具木球	P_Ball_Wood	道具石球	P_Ball_Stone
生命球	P_Extra_Life	分数球	P_Extra_Point
活式木栅栏	P_Modul_01	升降器	P_Modul_03
秋千	P_Modul_08	单摆	P_Modul_17
风扇	P_Modul_18	双向挡板(二翼桥)	P_Modul_19
推板	P_Modul_25	沙袋	P_Modul_26
软木桥	P_Modul_29	跷跷板	P_Modul_30
移动路块垫箱子	P_Modul_34	T 型板	P_Modul_37
箱式浮板	P_Modul_41	半球型钢钟	P_Dome
箱子	P_Box		

特殊物件组: 小节盘点火 PC_Checkpoints 开头盘点火 PS_Levelstart

重生点 PR_Resetpoints 飞船 PE_Levelende

小节组: Sector_0X (表示第 X 小节,飞船会出现在对应小节组数的那个小节)

影子组: Shadow *[具有影子坐标的物件归这个组后球在上面有影子,关于影子,此处仅需了解即可,之后会介绍]*

声音组: 撞击普通路面: Sound_HitID_01 撞击木板: Sound_HitID_02

撞击钢轨：Sound_HitID_03

在普通路面上滚动：Sound_RollID_01

在木板上滚动：Sound_RollID_02

在钢轨上滚动：Sound_RollID_03

注：

1.每个地图中归 Floor_Stopper 组的物件只有其中的第一个才有碰撞声音效果，直接纯 VT 制图暂时无法改变这一局面。


2.特殊物件组的物件只需要归特殊物件组即可，不需要归任何其他组。但对应特殊物件的命名一定要正确，否则很可能导致地图崩溃（游戏脚本通过审查地图中这些的命名而进行渲染）！如下是这四种物件的命名规则：

终点飞船的命名一定要为 PE_Balloon_01

重生点的命名为 PR_Resetpoint_0X（X 表示第 X 小节，比如第三小节的重生点命名为 PR_Resetpoint_03）

开头盘点火的命名一定要为 PS_FourFlames_01

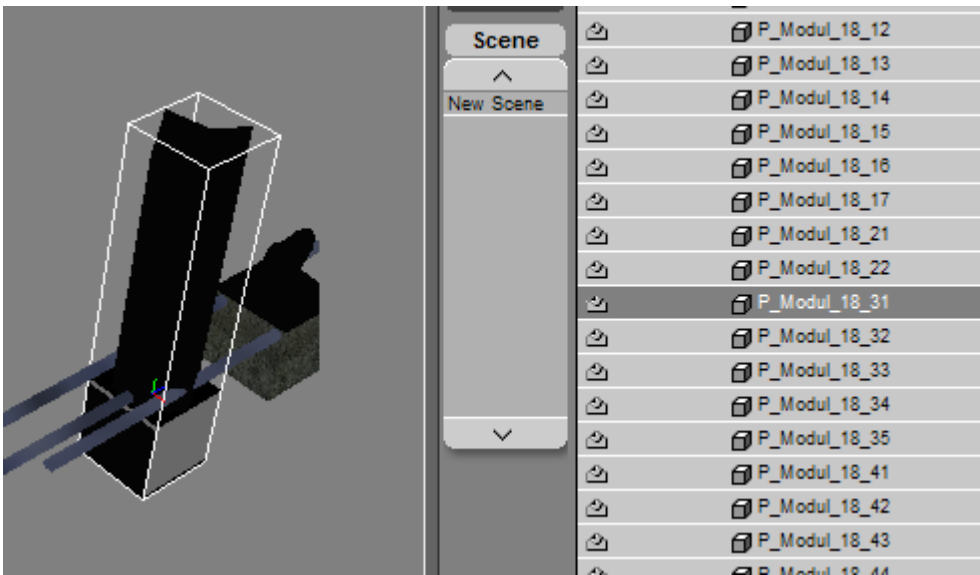
小节盘点火的命名为 PC_TwoFlames_0X（注意这里的 X 不表示第 X 小节，而表示第 X 个小节盘点火，因为第一个小节盘点火在第二小节，所以第二小节的小节盘点火命名为 PC_TwoFlames_01，以此类推，第五小节的小节盘点火命名为 PC_TwoFlames_04）。

3.如果在做的图中找不到需要归的组，可以点击 Level Manager 内左上角的图标来创建一个新组，或者在物件列表中对要归组的物件点右键，选择 Place Selected in New Group，这样就自动把该物件归到一个新组里。值得注意的是，新建的组需要右键 Rename 改名为对应要归的组名，否则对于 Ballance 地图是无效的。

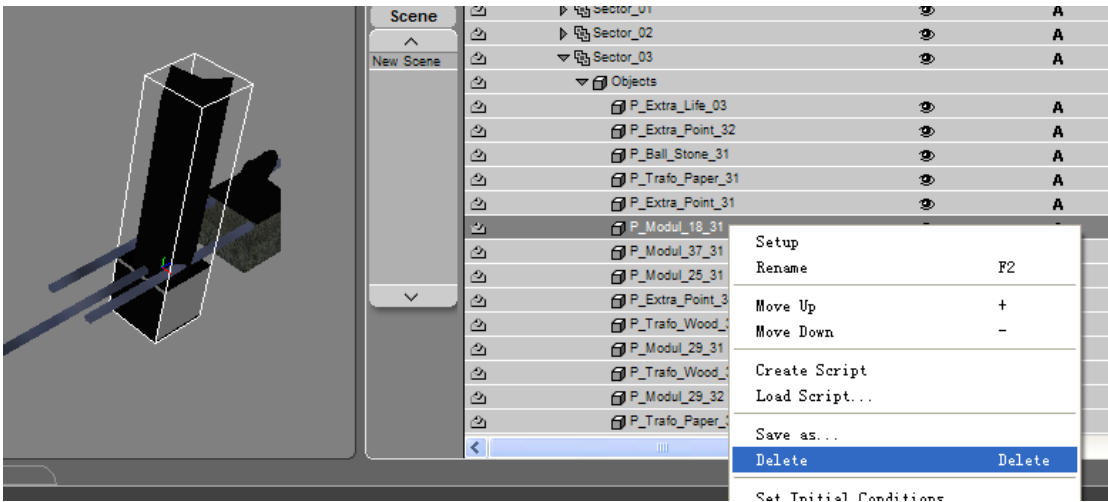
2.3 小节组改换

在上一节中我们了解了物件的归组，现在我们知道机关需要归小节组，并且在游戏中也知道，机关只能在归组对应小节发挥作用，而一玩到其他小节，原先的机关彻底消失（变球器可能消失，也可能是被隐藏）。（至于一个机关同时归两个或多个小节组，也只能在其中一个小节发挥作用）。所以在编辑或修改地图的过程中，想把其他小节的机关放到当前编辑的小节，需要进行小节组改换，通俗地讲就是把机关本来所在小节组的位置删掉，然后重新归到需要的小节组（此过程中容易出问题，如果小节组忘了重新归组，那么游戏将会打不开，后续核查是很困难的，所以用 **Save Version** 备份保存 **cmo** 格式地图是最好的选择）。

首先，我们应该先找到需要进行小节组改换的机关，并且需要明确知道其所在小节。当找到需要进行小节组改换的机关时，我们应该记住其名称。如图所示，我们想把某地图中第三小节的一个风扇重新归组到第四小节，以使其在第四小节发挥作用。（改换后不能再在第三节使用）

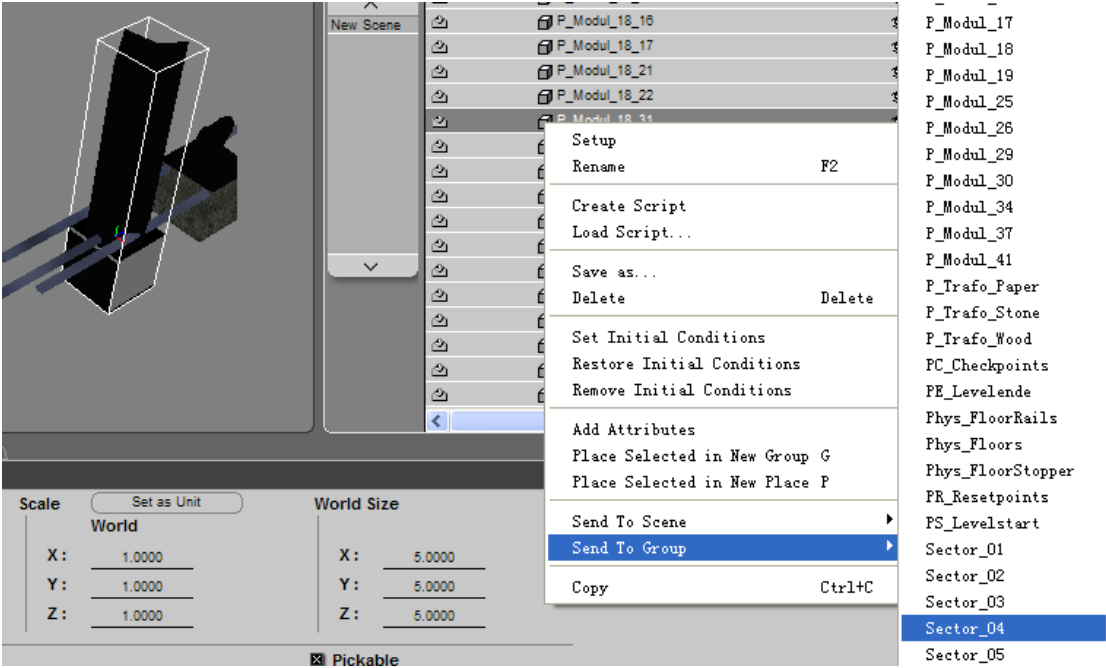


我们需要先在物件列表里找到这个机关物件，并且知道其名称是 **P_Modul_18_31**。我们需要暂时记一下这个名称。之后我们按照名称在 **Level Manager/Global/Groups/Sector_03/Objects** 中找一找这个物件。我们可以找到该物件，点击一下，发现 3D 框中该物件被选取了，之后右键归到第四小节对应组，



此时该物体同时归了两个小节组，必须在第三小节组内（Group/Sector_03 中）删除，以删除该物件在第三小节组的位置。

注意这里的删除是只删除该物件在组里的位置，而不涉及任何牵连，也就是物件及其所关联材质、贴图和网格都不会被删除。(同样的道理，想删除某个物件在组里的位置，也应该使用如上方法)



这样，我们就可以保存地图了。在打开这个地图时，就会发现该风扇在第四小节中运转了。（如果不删除物件在第三小节组的位置而直接归第四小节组，会导致之前归的组占“竞争优势”而使该机关仍只在第三小节运行，还有可能导致游戏崩溃）

类似的，道具球、分数球和生命球，乃至箱子，都可以这样小节组改换，有时候我们还可能会因为各种制图需求而对一些其他物件进行换组操作。

第三章 VT 元素概述

我们在上一章了解了物件和组，但是我们知道还有网格、材质和贴图这样的元素。VT 制图需要注重整体性，对元素的综合理解和应用是 VT 制图的基本要求。我们将在本章了解其他的 VT 元素，并且根据这些元素获得一个整体的制图方向。

3.1 网格概述

在之前我们已经了解到了网格（Meshes）。网格是描述物体形状的层面，被物件直接关联。物件本身不储存形状信息，因此如果你把物件的网格换成其他物件的网格，就会导致物体的形状变成其他物件的网格所存储的形状。一般情况下一个物件对应一个网格，也有多对一的情况，而一对多在静态地图中无意义，因为在同一时刻只有一个网格能被激活。

机关可以没有网格，因为机关是通过机关组的自范性（游戏脚本通过审查归机关组的性质来渲染）来体现性状的，在制图中机关的网格只是方便参考用的。*[事实上，机关可以使用任何网格，但其他使用此网格乃至其所用的材质和贴图都会丢失，因此不推荐随意篡改机关的网格。此现象的原理详见第五章第 1 节。]*

我们在制图时对网格的建议是：机关和重生点等特殊脚本渲染物件应该共用网格，而路面、钢轨和死亡区尽量关联单个网格（复制物件时顺带网格就是很好的），尽量做到“不共用”。虽然网格是地图文件大小的主要担当，但路面、钢轨和死亡区关联网格共用会带来不少问题。如下是共用一个网格时的危害：

1.对路面或钢轨进行放缩时，在游戏里会出现碰撞箱问题，即“假路面”及“隐形路面”。举个例子，两个钢轨共用一个网格时，如果对其中一个实施缩放操作，另外一个也会跟着缩放。*[此过程省略了一个重置网格的操作按钮问题，所以实际上描述并不准确，详见本书第四章第 1 节。]*

2.复制死亡区时网格共用同样会导致缩放问题，一旦缩放失误，很容易造成死亡区与关卡区域重叠或关卡区域下方缺失死亡区有效部位的问题。

3.导致球的影子重叠（特别是风扇底座）。即若多个共用同一个网格的正常路面都归了影子组，那么球在走上其中一个时所有的路面上面对应的位置都会有影子出现。

[路面没有影子的问题不是网格问题，而是物件问题。只有原版地图的物件才有影子坐标，要想使某些自制的路面也有影子，只有保留原版地图路面的物件，再赋予它我们自制路面的网格才行。因此建议如果要大量复制单独的路面，可以先把第一块路面加上影子后，复制时把网格也连带复制。（理解这段话可能较难，但没关系，本书第四章的第 1 节和第 2 节会进一步讨论这方面的问题）]

我们可以在 Level Manager/Global/Meshes 中找到网格。在双击物件进入物件属性时左下角的框中也会显示该物件所关联的网格。我们可以双击点开一个网格来剖析其设置窗口，如图所示：



网格设置里需掌握的参数是 Lit Mode，如果从 VT 之外导入的网格是 Prelit（很明显，因为在 3D 框中调用 Prelit 网格的物件会高亮），就需要在网格设置中改成 Lit。另外，可在网格设置里看到调用该网格的物件和该网格使用的材质。

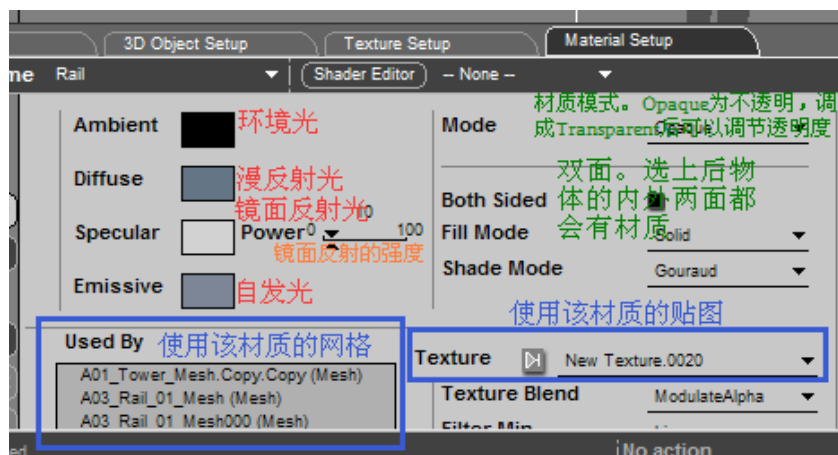
3.2 材质与贴图

我们已在前面接触材质（Materials）和贴图（Textures）。Ballance 的网格一般除了死亡区外，都需要有视觉样式的需求，而 Ballance 的视觉效果是材质和贴图共同作用的，到目前我们就来仔细分析一下材质和贴图的概念及作用。

Ballance 的材质和贴图都采取的是最简单的创建方式——好比用一张纸贴到物件上。材质一般只在关卡内存在，且只占很小的硬盘空间，而贴图可以分内置贴图和外置贴图。在 Level Manager/Global/Textures 里双击点开某个贴图后如果什么都没有，就是外置贴图，需要在外部（Ballance/Textures 里）寻找有关文件；相反，如果是内置贴图就会有一张图片，该图片所占用的硬盘空间就要由这个关卡来承担了。

在 VT 界面中，我们可以通过左侧工具栏的  来创建材质（左）和贴图（右）。创建或在 Level Manager/Global/Materials 中双击材质可以找到材质属性，如图所示：

此图片展示了材质属性中很多有用的设置，根据图片上的文字解说灵活运用，对材质的把控就能来去自如。

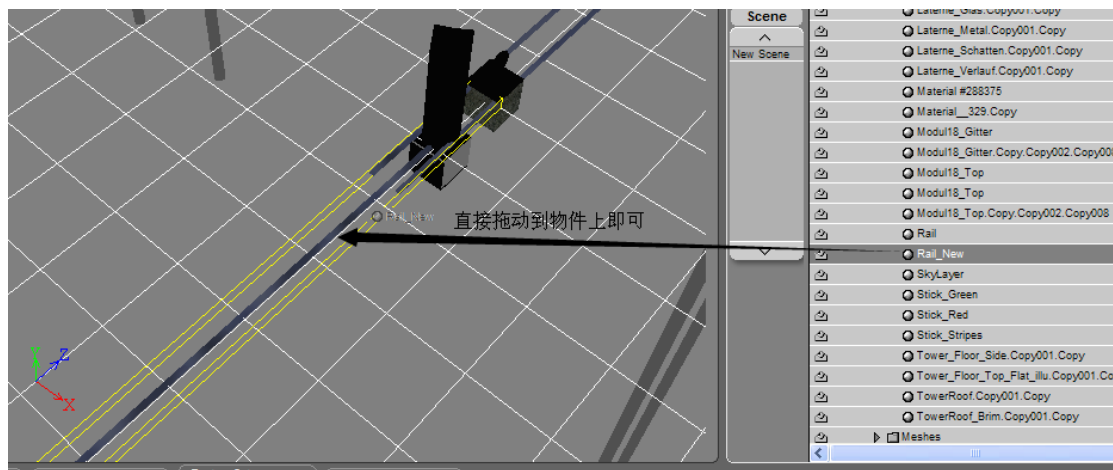


材质可以关联其对应使用贴图，这样将材质应用至物件就会是材质和对应贴图协同作用的效果（如果材质所关联的是外置贴图，那么在 VT 编辑界面只会看见材质的颜色，而在游戏中物件的外观却是材质和贴图协同作用的效果）。材质也可以不关联贴图（即图中右边蓝框中是 None），这样在 VT 中使用材质的物件只会显示材质的颜色。

与材质同理，新建或进入贴图设置也可以看见如下菜单：



将材质和贴图应用到物件上很简单，只需要把对应材质从列表中拖到 3D 框里的对应物件即可（严格地说，拖动材质到物件上其实是改变物件调用网格的材质的应用，所以其实是应用到网格上，因此如果存在共用该网格的物件，那么那个物件的材质也会被改动。拖动贴图也只是改变材质所关联的贴图而已），物体的外观就会是材质和材质所关联的贴图的作用结果。

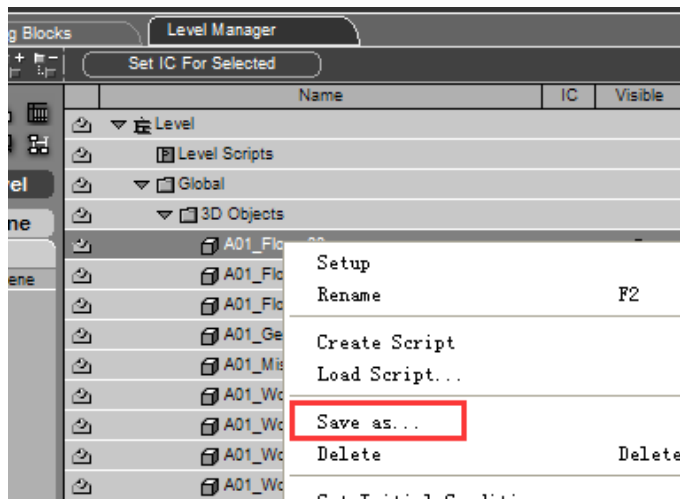


至此，我们可以得出 VT 制图元素的关联关系：物件关联网格，网格关联材质，材质关联贴图。对于贴图的内外置问题，较为复杂，此处不予介绍，详见本书第四章第 3 节。

3.3 素材的导入与导出

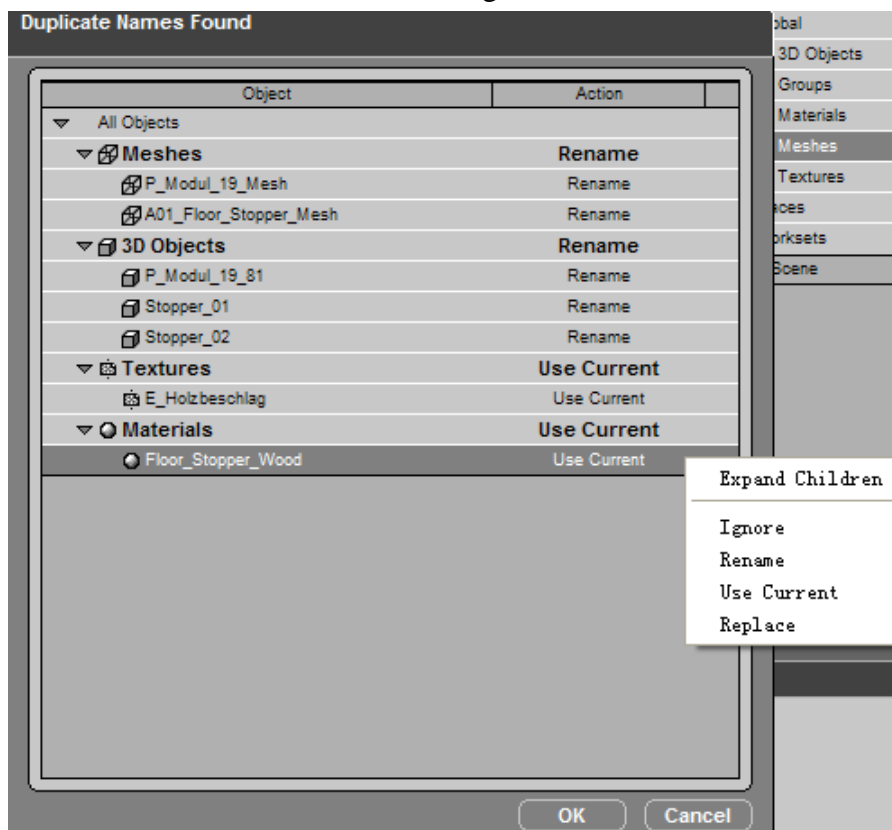
我们早在本书第一章第 2 节就已经了解到了素材。学过了物件及其各种从属关联后,我们可以来进一步讨论素材这个问题了。素材可能来自一些地图的部件,当我们在一些地图中发现有用的物件后,可以对该物件点右键,选择“Save As...”另存为 nmo 素材(不过导入的物件坐标会在导出时的位置),下次使用即可导入。我们现在再回顾一下当时直接在 level 上右键“Save as...”不就是把整个关卡导出成 nmo 吗?

当然,有些素材是用 3ds Max 等软件通过建模做出来的,不是从地图中提取的。



我们在导入素材时仍然用的是 VT 打开 nmo 格式地图的方法。即用 Import File 来导入。导入素材文件有时候会引起撞名(名称和地图本身部分元素相同),这时候系统就会给出窗口,让你选择撞名元素的名称及解决方案,如图所示:

右键每一项的 Action 都有 4 个选项(Expand Children 对解决该问题并没有什么用,所以不考虑),本来默认为 Ignore(忽略),即忽略撞名,直接强行导入元素,不支持选择 Ignore 来处理撞名问题,名称易混淆而且很可能导致游戏



出错。Rename 是重命名(会加上数字 001、002 等以区分),由于网格应该不共用,物件是单个个体,所以物件和网格一般应选用 Rename。Use Current 是使用场景中现有的,舍弃新导入的。一般建议材质和贴图撞名时用 Use Current,这样对于内置贴图,可以节省硬盘空间。Replace 是用新导入的替换掉场景中现有的,一般建议在纯 VT 制图中不要使用 Replace,万一替换掉现有重要元素就不好了。但

理解 **Replace** 选项的意义是很重要的，如果之后需要通过 3ds Max 这样的建模软件来修改素材，可能会更灵活的运用素材导入撞名解决方案，如地图中的一个复杂钢轨之前就是用 3ds Max 做出来的，并且已归好组，那么用 3ds Max 修改此钢轨并导出素材，然后用 VT 导入时，就可以给网格选 **Replace**，而物件、材质和贴图都选 **Use Current**，这样相当于是仅替换了形状而保留正常的原有材质贴图，已归好组的物件也不需要重新归组。

素材的导出方法就是从 **Level Manager** 列表中将目标文件以右键 **Save As...** 的方法导出。一般可以按住 **Ctrl**，按照需求选择多个元素再导出（导出的元素所关联的其他元素也会一并导出，但是最常用的导出物件只会导出所关联的网格、材质和贴图，而不会把组导出，所以导入的素材都是需要重新归组的）。


附：死亡区改造法（概念提出者为 **BallanceBug**）：对于纯 VT 制图者来说，由于没有 3ds Max 的强大制造网格的功能，制造大规模的路面往往很难。为了造出具有一定水平的路面，可以运用如下方法（注意：此法无法获得原版材质路面，但是可以做到原版路面的影子、碰撞声音和滚动声音）：

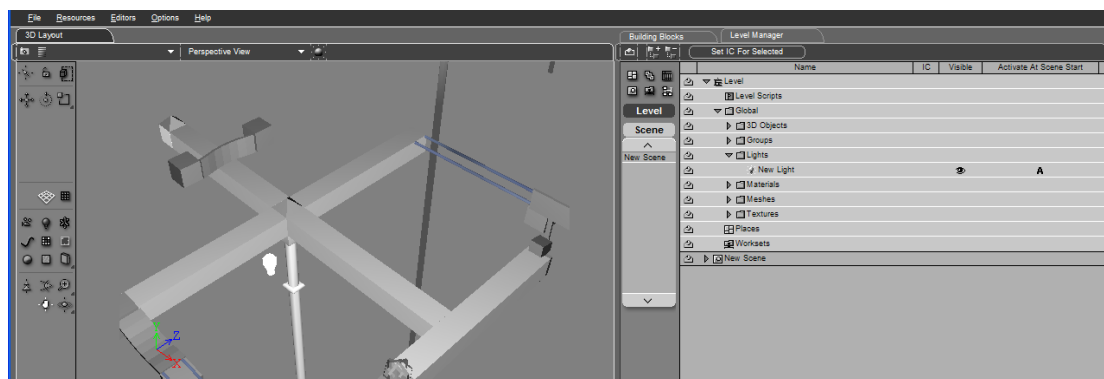
路面的单位是一个长方体，而原版地图中常见的长方体是箱子和死亡区，由于死亡区好找，建议使用死亡区改造。把死亡区复制（需要连带着网格一起复制）后把复制物名字改成需要的命名（做到令人不误解即可），从死亡区的组里将复制物删除，让复制物显形，并归到路面组、影子组、路面撞击声音组、路面滚动声音组。然后在物件设置的第四栏里调整大小（**World Size**, VT 中的 5x5x5 相当于 **Ballance** 里的 1x1x1 路面），调整完大小后点击物件设置第三栏上面的“**Set as Unit**”按钮，一定要注意不能共用网格。如果要给路面上色（不上色会是灰的），可以用 VT 中的创建新材质功能，把材质的颜色调一下然后应用到这个物体上。也可以再增加贴图，但是增加原版贴图是无法取得原版效果的，会发生贴图错乱。

再次提示：材质对应的是网格而不是物件，调整单独一个物件的材质会导致其它相同网格的物件的材质也发生变化，所以区分网格极其重要。

至此，我们已经将可供 VT 制图的基础内容全部学完，不过我们需要认真的态度、活跃的思维和不断的坚持才能做好一张自制地图（参见本书第五章第 2 节）。之后的章节中会更深层次地讨论 VT 的有关内容，建议有兴趣制图的继续阅读。

3.4 光源简介

我们在第一章第3节中就简单的了解了一下“灯泡”，我们将在这里进一步了解光源。作为一个独立的模块，光源与物件、组、网格、材质和贴图的不同之处就是原版地图中没有光源，用创建灯泡或者用 VT 打开存在光源的自制地图后会在 Level Manager 里出现新的一个目录“Lights”，同时我们发现有了灯泡周围的环境照亮了，如图所示：



其实光源分为三种，最常用的是平行光源。平行光源是光线方向平行的光源，对有着固定的照亮方向。

默认创建的灯泡是点光源，有一定的照亮范围。也可以在创建光源后手动设置。我们可以在 Level Manager/Global/Lights 中双击新建的灯泡进入光源设置，如图所示，通过调节各种参数来设置点光源的属性：

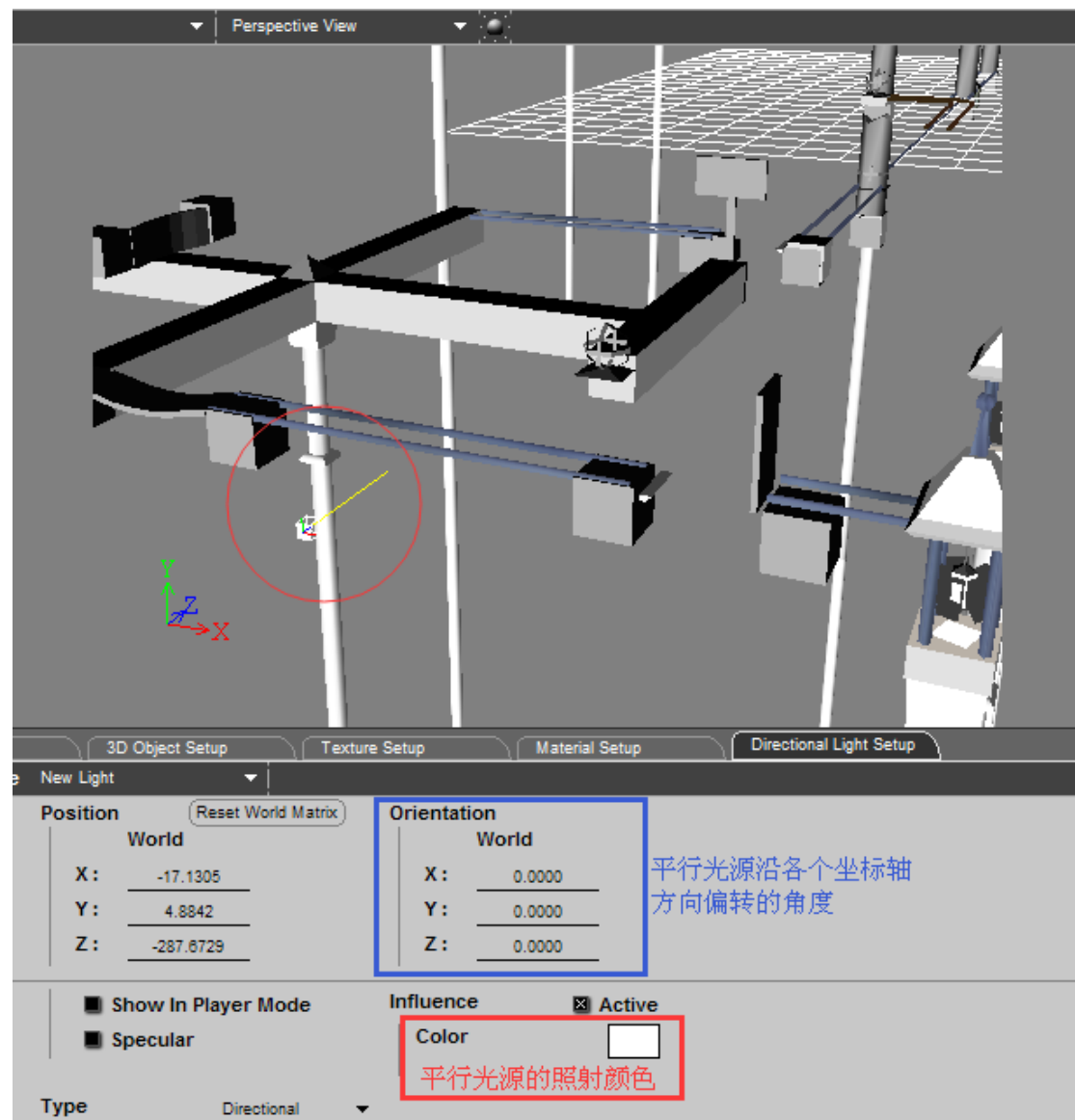


我们可以在左下角“Type”栏调节光源种类，Point 是点光源，Spot 是以投影仪形式的范围照射性光源（可以用“手电筒”或“聚光灯”来理解，可以调控的范围参数较多，具有灵活性。Spot 型光源在创意地图中可以营造效果，也可以应用于某些局部照亮），而 Directional 是平行光源。平行光源是很有用的，可以从某一方方向照射全部物件，而且在部分地图中，平行光源可能还会作为地图灯光。

我们将该光源切换为平行光源，现象如下图所示，我们会发现光源的样子变成了一个平面和一条射线的形状，而且整个关卡范围的物件都受到了平行光照。

我们在设置框中还会发现光源出现了调节物件的旋转角度的属性框（参见下图中的蓝色框），可以调整平行光源发射光照的角度。

另外，调节光源照射的颜色框还在，但是却没了光照范围，因为平行光源属于全范围平行照射。还有一个特点就是平行光源无法调节亮度，平行光源的亮度恒定为 1 单位（点光源可以在 0~10 单位 之间任意调节）。目前尚未能总结出平行光源不能调节亮度的原因。不过我们可以把平行光源的照射颜色调为灰色，从而间接降低其照射的亮度。



如果编辑地图后保留光源而不删除，在游戏中我们将看到光源波及的地方有额外的照亮，这原则上不符合 Ballance 默认关卡的逻辑，但已有部分自制地图标新立异，利用装饰性平行光源给地图营造了别样的光彩。

光源勾选 ☒ Specular 时会显示出镜面反射，从而让钢轨之类镜面反射率高的物件更加光亮。

删除光源不能利用 Options（选项）——Unused Objects Explorer 里的删除，需要手动在 3D 框里找到光源删除，或者在 Level Manager 的列表中删除。由于光源相对独立，不与物件、网格、材质和贴图相关联，所以删除时既可以选择 No Dependencies，又可以选择 Full Dependencies。但是为了防止在其他时刻删除物件习惯性误用 Full Dependencies 而导致的从属删除问题，在删除灯泡时还是推荐选择 No Dependencies。

第四章 VT 元素拓展

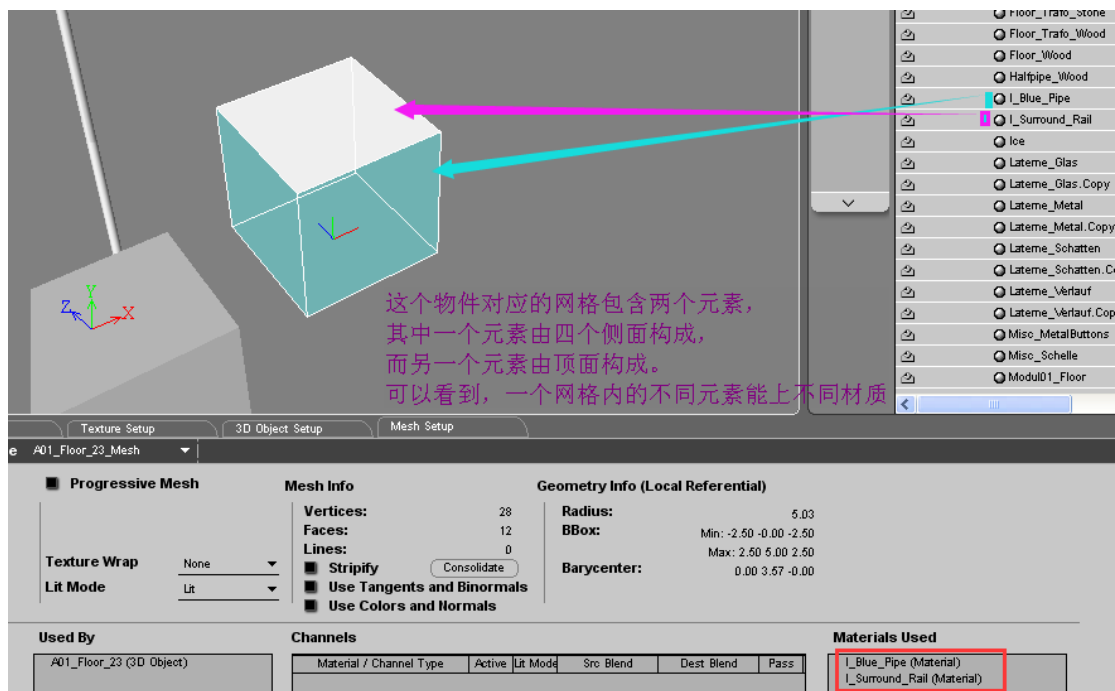
VT 制图元素的整体掌握需要更深层次的理解才能贯通制图，而在这之下衍生出来的一些制图技术操作常常为人们所用。本章将进一步深化理解，并根据对 VT 元素的进一步分析提出一些对 VT 制图有用的方法。

4.1 用什么表达 3D

在计算机的世界中，点是构成空间的基本要素。三个点可以确定一个平面（该面只有一边有法线，也就是说，从某一边看该面是正常的，而从另外一边看就是黑色的，在 VT 中该面的视觉效果是可穿透的，例如导入原版关卡后从下往上看）。当然如果所用的材质勾选了双面材质，就会在两面都看见。当若干个面围成一个封闭几何体（黑色的面均朝里，正常的面均朝外），就形成了一个基本的几何体。（注意在三维设计软件中没有完美的曲线，一个曲线是由许多个首尾相连的线段构成的，这与物理学中微元法化曲为直的思想是相似的）。

在 VT 中，实际描述这些几何体的是网格（Meshes）而不是物件（3D Objects）。物件是网格的载体，本身几乎不占存储空间，而描述三维参数的网格才是主要占存储空间的元素。VT 用这种方法将物件和网格分开，其目的是让网格作为可行的通用标准物，而物件，可以直接调用网格来达到自己所需的形状（所以无网格的物件在 VT 中只会显示出坐标轴，而无任何形状）。这样需要完全相同形状的物件就可以共享同一个网格，不用再在每次建设物件的时候单独描述其形状，这样也省了物件所占的硬盘空间（路面影子问题是游戏里的特殊设定导致的，所以相同路面为了防止影子重复而网格不能共用，在下一节中我们会了解到）。

我们在第三章中了解过，网格可以使用多种材质。材质和贴图共同作用可以



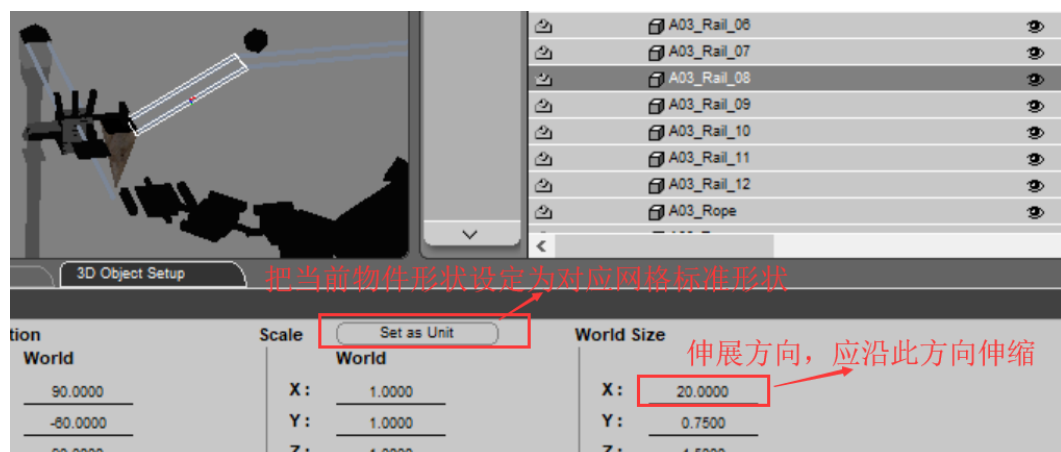
修饰一个网格的外观。而多种材质在同一个网格的实现，就是分别在不同元素上的实现。一个网格中，单个或多个面可以构成一个元素（注意：这里的“元素”是指网格中作为上材质单位的元素，并非 VT 制图元素，在理解时应把这两个概念区分），而元素组成了网格，也就是说虽然网格是一个整体，但可以分别往网格的不同元素上上材质（将材质拖入 3D 框中的物件以上材质的本质其实是给物件使用的网格上材质），当然只包含一个元素的网格（如没有附加螺丝钉的钢轨）就只能上一种材质。

当导入的素材物件的材质存在缺陷（变暗或材质错乱）时，我们把正常的材质拖动到其材质缺陷的元素上即可修补存在缺陷的材质。

而需要注意的是，如果一个网格所包含的两个或多个元素在 VT 中被赋予同一个材质，那么它们将变为同一个元素，再赋予材质时会按照一个元素来处理，所以就只能用一个材质了。（猜测网格的材质元素与其材质方式有关，因为网格不仅存储形状信息，还存储材质方式信息，当两个材质元素被合并后自然就只能容纳油管材质了）

我们这里再提出一个对于网格原理的应用。由于归钢轨组会使材质自动平滑，所以对直钢轨的伸展方向进行伸缩，几乎造成不了影响，因此直钢轨沿伸展方向的伸缩对制图具有很大用处。事实上，由于路面不能放缩（Ballance 内的影子是由路面而不是球决定的，根据放缩状况不同，影子可能出现单轴拉长或过大过小的问题），在精细一些的纯 VT 制图中，为了弥补无法建模而导致一些部位不能完美对接的问题，一般钢轨的伸缩和角度调整就成了方便的调整方法。然而，伸缩也会牵扯到一系列问题，因此本书第二章第 1 节提到物件设置介绍时就已经有了伸缩一项，但是没有详细介绍。

按照物件与网格的原理分析，其实对物件进行伸缩时，其实只是改变物件调用网格的配比，储存物件本身形状的网格并没有发生变化。因此，这会导致严重的碰撞箱问题。比如把一段短的直双轨拉长，然后再旋转一个坡度，实测时就发现有效碰撞部位明显不是这样的。为了解决这个问题，我们只能使网格的形状也是这样。所幸，这只需要按物件设置中的一个键（Set as Unit）即可。



因为我们在伸缩、调整角度后按 **Set as Unit**，就会把网格的标准设成这样，从而简洁地改变了网格。当然这就会解决碰撞箱问题，使对应的钢轨能用。

不过，由于这是设置网格标准，如果该钢轨有共用的网格，那么共用这个网格的物件也会发生同等程度的伸缩，给游戏体验造成不便。因此，强烈不推荐直钢轨共用网格。（如果已有不能避免修改的网格已经共用的情况，需要互相独立可以参考下一节的末尾部分）

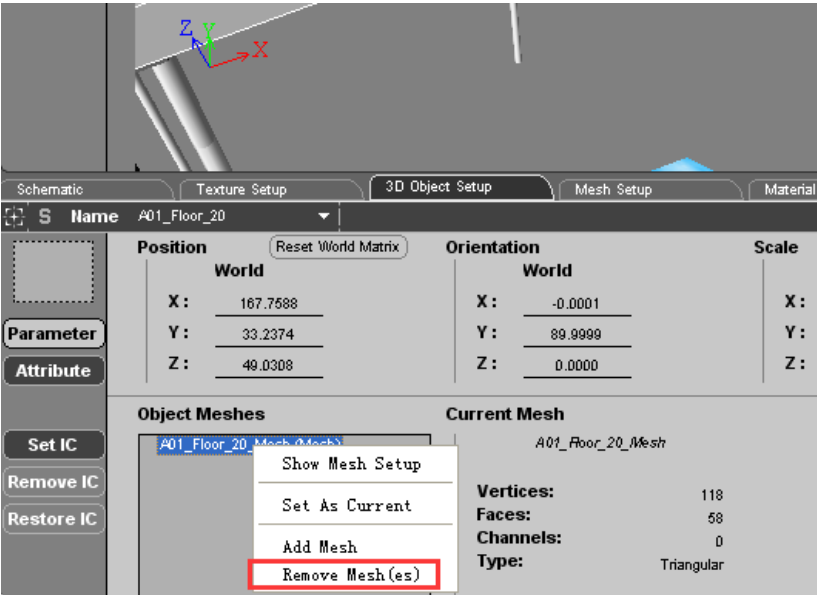
4.2 路面加影子

我们在本书第三章第 1 节中了解到球在路面上的影子效果，并且知道这和物件的属性有关。网格只是一个给物件描述形状的存在，而物件的一些其他属性，如坐标和选转角，是由自己来描述的。在操作 **Ballance** 地图的方面，我们需要了解的物件自身属性还有影子坐标。当拥有了影子坐标的物件归了影子组，才会使水平位置与该物件相同且竖直位置相差不大的玩家球，在该物件上形成影子。

用 **3ds Max** 制作出来的物件，是不存在影子坐标的，所以无法形成影子，我们需要通过给物件加影子来实现影子效果。

影子坐标是个难以自主设定的存在，它可能涉及非常复杂的步骤，也可能是游戏的特殊设定才造成的，所以我们加影子显然不能对物件进行操作。但是，原版地图中的物件是现成的具有影子坐标的物件，我们可以取原版地图中的物件，来替换我们本来具有想要的形状（网格）的物件。（所以在第三章中说没有影子的问题是物件的问题）。加影子的本质是物件的替换，而不是网格（仅描述形状）的替换。

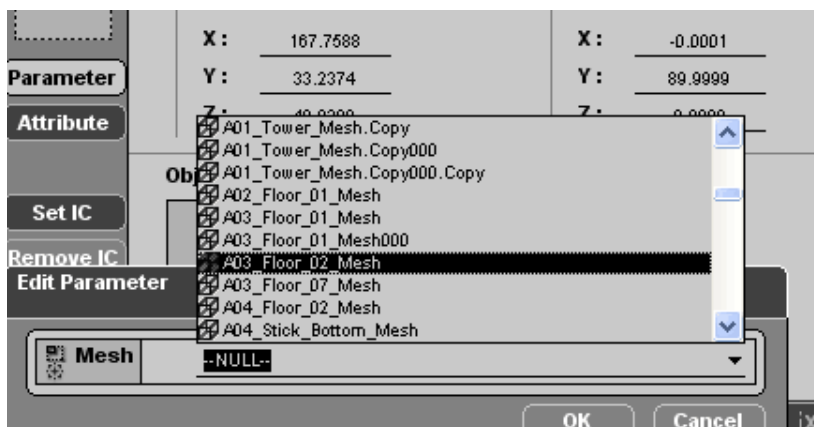
当我们找到一个原版的路面物件，双击进入该物件的设置，会发现该物件使用了一个描述自己形状的网格，我们右键那个网格，点击“**Remove Mesh(es)**”（或者选中网格按 **Delete** 键，这样不会删除网格但会删除物件与该网格的调用关系），即可使该物件原有的网格被取消。取消之后我们会发现该物件“消失了”，但其实并没有，这只是无网格的物件的特征。在 **Level Manager**



中找到这个物件，可以在 3D 框中看见原先物件的位置出现坐标轴图案。

此时找到我们需要加影子的物件，双击进入物件设置窗口，找到该物件对应的网格，记住网格名称。再回到具有影子坐标的路面的设置，同样在 Object Meshes 框中右键“Add Mesh”，选择需要加影子的物件对应的网格，该有影子坐标的物件即显现出需要加影子的物件的形状，此时这个物件就是最后“成功加了影子”的物件。通过移动物件的方法，使这两个形状（网格）完全相同的物件重合，再删除原本无影子的物件。此时路面加影子的操作就完成了。

具有影子坐标的路面，归了影子组后显示出影子，但并不是取玩家球在路面上的正投影。事实上，只要玩家球与路面在竖直方向上相隔距离不大，而水平位置相同，可以在路面含有元素的任意竖直位置出现影子（可以使用弹力球来测试，当玩家球与路面的高度差较大时影子即消失，另外球在路面长方体内也会使路面上出现影子）。所以在同一水平位置具有多层的路面，乃至顶面为标准路面、下方为倒悬的四棱锥型的浮空岛路面，加了影子后会出现影子重复投映和假影子的问题。另外由于影子坐标的固定性，将一个具有影子坐标的路面沿水平方向伸缩后，会导致影子拉长或缩短的问题。共用同一个网格的不同路面若都归了影子组，由于影子坐标体现在同一网格，则会出现影子重复。



对于影子伸缩和影子重复的问题，只要我们留意，不随意伸缩物件，也在复制路面文件时尽量复制关联网格，这种问题就不会出现。但同一路面不同高度出现影子是个令人头疼的问题。对于同一水平位置具有多个层的路面，暂时没有有效的办法解决问题，我们可以不加影子，因为不加影子的效果可能更好。对于浮空岛型路面，我们可以在 3dsmax 创建它（这里不作讨论，需要学习 3dsmax 建模的可以找专门的教程）的时候，就将其顶面与倒悬锥体分离，让顶面归影子组，而倒悬锥体不归影子组（或者“不加”影子坐标）。这样我们就可以得到上面具有影子，而又不投映到锥体上的浮空岛路面了，



如上图所示。

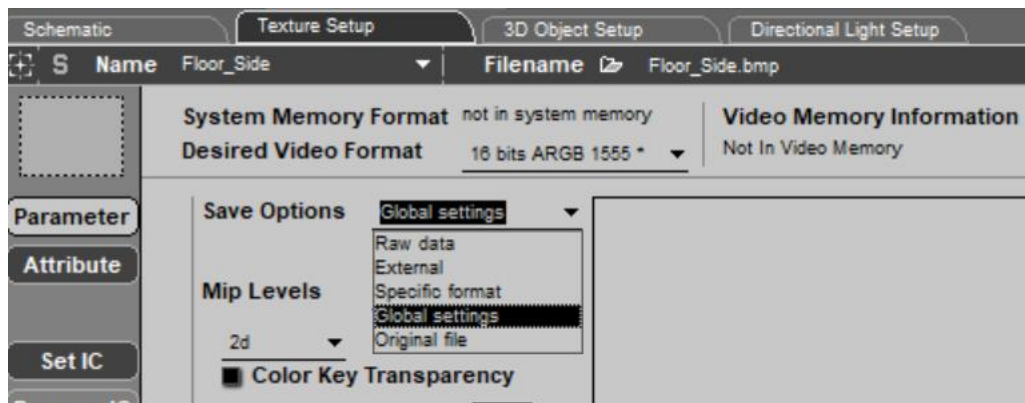
附（上一节中已不慎共用了网格而不得不改的情况下如何解决）：不要着急，虽然比较麻烦，但是可以做到。首先我们要找到对应的网格，然后单独复制网格，选择无从属（No Dependencies）（这样是为了共用材质和贴图，节省地图大小），之后按照本节删除物件从属的方法把共用网格的物件与网格的关系删除，然后添加复制后的网格。这样两个共用物件的网格就分开了。其实，理解熟练了就可以对 VT 的各种操作融会贯通。

4.3 贴图的内置和外置问题

对于很多地图，如果是在编辑的 CMO 格式，那么放在 Ballance/Textures 里之后，含原版贴图的元素都会显示贴图（其实就是内置了），同时因为关卡承担了贴图的文件，所占大小会增大很多。当想把这样的地图改回来时，却已发现力不从心。因此很多人愿意以此方式在 vt 中预览自己编辑的地图但却从不敢在那里保存地图。不过，我们可以详细了解一下贴图的存储机制，就能有效应用了。

贴图的内置和外置问题确实是一个让人头疼的问题，虽然内置保证了直观性，但是改回外置却很麻烦。我们可以先了解一下 VT 编辑地图中贴图存储的机制来寻找对应的办法。

我们在 Level Manager/Global/Textures 中点开贴图设置，会找到贴图的存储格式设定框。如图，在那里调的都是贴图的存储格式，我们只需要明白其中的三种：Raw Data(内置贴图)、External(外置贴图)、Global settings(服从默认设置)。

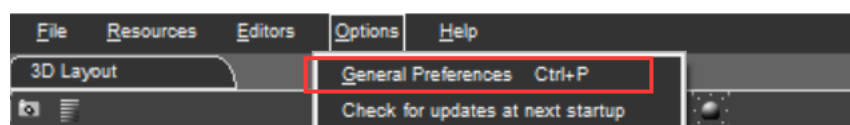


那么这样不是很简单吗，如果想把单个贴图调内置贴图，就把这项设成 Raw Data，外置就设成 External 就行了。然而，真正的问题并不是如此，提出了问题，知道问题的走向，才知道解决这个问题相对比较棘手。

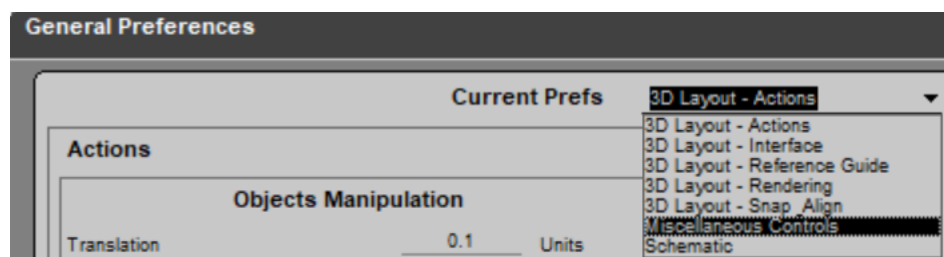
我们先了解一下第三个要说的贴图存储方式：Global settings，什么叫服从默认设置呢？就是 VT 有一个对很多参数的默认设置（General Preference），其中就包括对贴图存储格式的设置，也分为内置贴图和外置贴图以及一些在制图中用不上的贴图存储格式的设置。如果地图的贴图设置选择了服从默认设置，那么就

会按照默认设置来走。而我们常用的 VT 默认设置里将贴图存储格式设为内置贴图，只不过由于缺少图源，所以会显示灰色，其实那是贴图缺失的表现（不过如果设置为外置贴图的贴图，改变原贴图的路径位置，也会使那个图片框内显示灰色），如果在与 CMO 地图文件同级的文件夹里有名称相同的贴图，就会被“吸”进去成为内置的贴图，这也解释了把 CMO 地图文件放在 Ballance/Textures 里之后原版贴图会全部变成内置的原因。

如果要给一张地图的所有贴图都变成外置，那么一个个调会造成巨大的工程量。所以我们可以调整默认设置中的贴图存储方式来使贴图变回外置。General Preference 可以在很多地方打开，当然最官方的打开方法是在 Options 中找，如图所示：

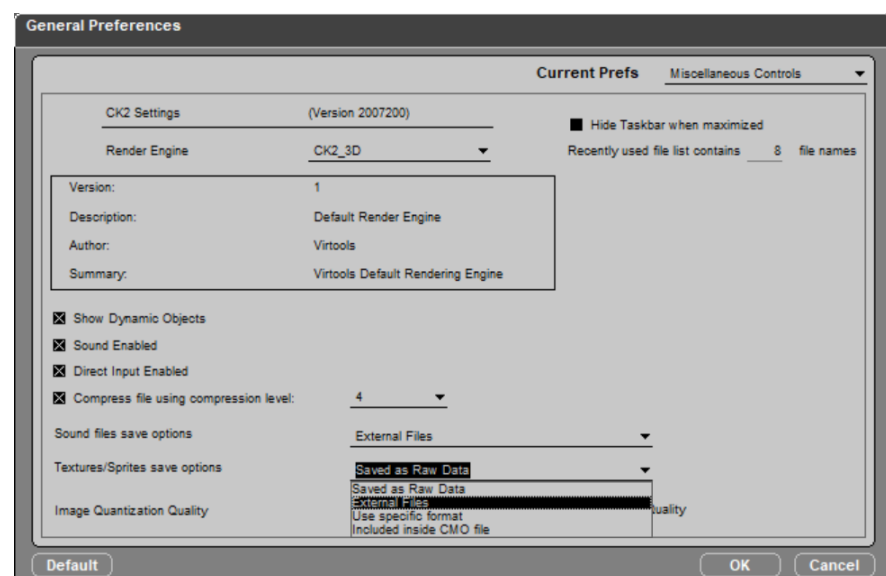


除此之外我们对左边工具栏的绝大多数工具图标点右键，进入的其实都是 General Preference 的不同方面。我们可以在 General Preference 的右上角找到不同方面的默认设置。贴图的存储格式在 Miscellaneous Controls 里，如图：



我们在这里可以把默认设置中的贴图存储方式改成外置贴图，就可以了。

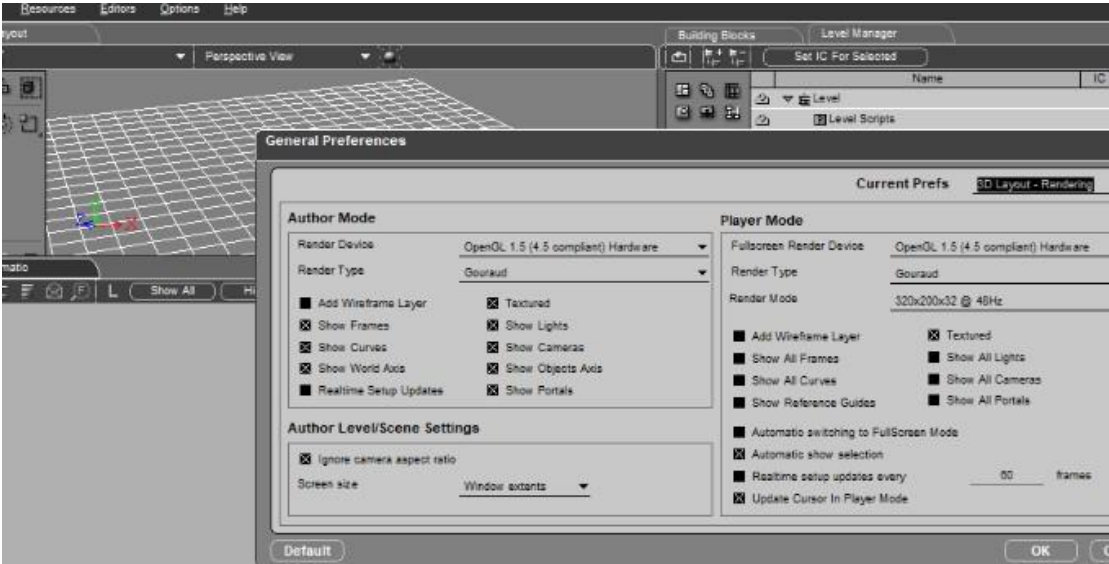
不过需要注意的是，这样的话某些加入的特殊元素或者自定义的非原版贴图必




须强行内置（贴图设置里 Save Options 里调为 Raw Data），不然在地图中将无法读取，从而只以材质的颜色等参数显示！因此如果要以这样的方式将内置原版贴

图都改为外置,那么使用一些自定义的非原版贴图(比如浮空岛路面的石锥贴图)时就需要在对应贴图的设置里将贴图存储格式设置为内置贴图了。

事实上, General Preference 还可以调整很多通用设置, 比如在 3D Layout - Rendering 里勾选 Ignore camera aspect ratio, 这样就可以使 3D 框完全显示里面的场景而不是只显示成比例的区域, 如图所示:



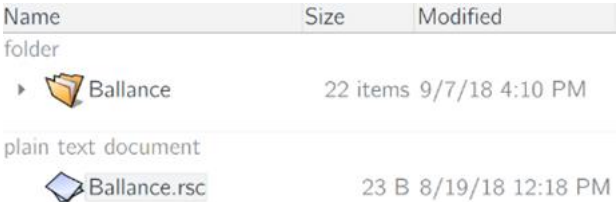
除此之外, General Preferences 里其实可以发现更多通用的设置。我们在第一章第 3 节调整单次旋转角度时, 右键出的窗口就是 General Preference/3D Layout - Snap _Align 里的选项。同样在  图标上右键也可以弹出 General Preference/3D Layout - Actions 里的选项, 其中主要可以调整镜头变换的属性, Translation(Pan) 是此图标拖动速率的选项设置, 建议设为 50, 这样可以更方便的在地图中拖动游览。在需要精确对齐时可以把它调低, 需要利用拖动镜头来快速切换视角位置时可以把它调到几百。总之, 灵活运用里面的一些设置, 对制图具有很大帮助。

4.4 RSC 目录的导入和应用

RSC 目录是指 VT 的资源目录, 我们可以通过把 Ballance 的根目录作为 VT 编辑的 RSC 目录, 这样就可以方便我们管理和整合资源, 也在一定程度上有更好的制图体验, 如下是操作步骤:

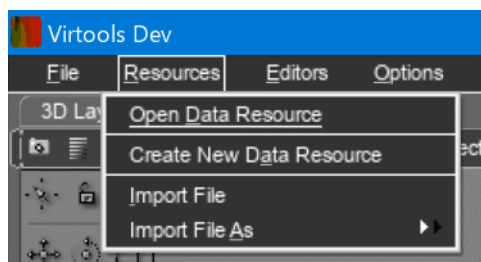
(1)在与你的 Ballance 安装目录同级的目录中创建一个文本文件, 在其中输入内容: Virtools RSC File 1.0

(2)将此文件重命名为<Ballance 安装目录名>.rsc。(假设你的 Ballance 安装目录名为'Ballance', 则应该创建 Ballance.rsc。)注意关闭「隐藏已知文件类型的扩展名」选项。下面的例子展示了完成该步之后的结果



(Ballance 文件夹内为 Ballance 的文件)。

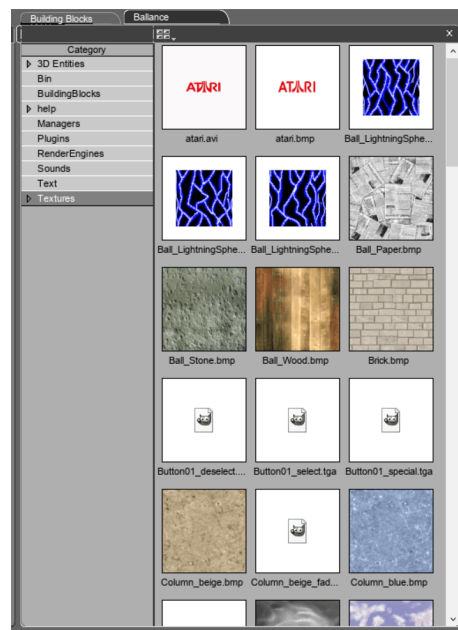
(3) 在 Virtools 中，选择 Resources – Open Data Resource。选定刚才的.rsc 文件。



(4) Virtools 中此时会打开一个新的标签，你可以直接使用拖放从 Ballance 的目录中导入资源。而且从此导入

Ballance 的 nmo 文件时，贴图都可以自动载入了。不过你的默认设置贴图如果还是内置贴图的话，可以通过 VT 的 General

Preference/Miscellaneous Controls 来改成外置。当然，此操作有时会发生自动变回内置贴图的情况，可以再次通过上述统一改外置的方法改成外置。导入 RSC 目录的效果如图：



第五章 游戏机制与制图

了解了 VT 制图元素的机制，我们还需要结合游戏机制进行总结，并根据一些现有制图技术来提升，以达到较好的制图效果。

5.1 对机关的总结

我们已经在之前的章节中讨论了机关这个概念。本节我们将给机关一个最终的定义和总结。

机关是 Ballance 的一种特有元素，游戏通过识别其归组和坐标轴的位置以及旋转角等参数信息来强行渲染，使其达到游戏中特殊的机关效果。真正描述机关性质的文件是 3D Entities/PH 里的 NMO 配置文件，那里才是机关真正用的网格，并且其能运动和只在当前小节激活的特点也是因为那些 NMO 配置文件含有脚本。

因此在 VT 制图中，机关的网格仅作参考，方便制图时选用。游戏会识别机关组并且识别小节组，如果归了机关组而未归小节组就会导致游戏崩溃而卡进度条。当游戏审查完并且确认该物件是机关后，就会把其本身的网格“切掉”，并且强行换用脚本来进行渲染。

经过现象的分析，游戏“切掉”机关参考网格的从属方式是全从属，也就是这个网格所用的材质和贴图也会被强行删除，所以永远不要将任何材质拖动给机关型物件对应的网格，否则会导致同样使用该材质的非机关型物件的网格失去材质样式，变成灰色！

分析了机关后我们不妨再分析一下审查机关的形式——组。其实，组是一个用于规划 VT 中元素的概念，因此不仅物件，网格、材质、贴图甚至组本身都可以归组。但是，这些归组操作对于制图的用处不大，甚至导致游戏崩溃，毕竟 Ballance 游戏的组机制是通过审查组的命名来赋予组内元素的物理属性。再次谈到归组的目的是巧妙利用机关的审查机制来让其他含有坐标轴的位置和旋转角等参数信息坐标和的场景元素变为机关。

我们知道，光源可以在作图时很好的照亮地图，方便看清，尤其是平行光源，因此我们可以放置一个光源来制图。但是一般情况下光源会给玩图时带来不是非常好的效果，所以很多制图作品在后期处理时都需要删除光源。不过由于光源含有位置坐标信息和旋转角信息，又可归组，所以可以利用机关形成的条件将其归机关组和对应该的小节组，让其在 VT 中显示为光源，而在游戏中实去光源效果变为机关！

所以如果把一个用于在制图时的平行光源归到一个机关组(和对应小节组),那么在游戏中就不会显示出其光源性质而会显示为所归机关组对应的机关。

不过,网格、材质和贴图没有位置坐标等信息,不满足构成机关的条件,所以不能归机关组,否则会导致游戏崩溃。

5.2 制作好一张地图

我们已经掌握了对 VT 的认识以及操作 Ballance 地图的很多知识。但做好一张地图,还需要精心的设计和合理的结构,我们将在这里讨论和总结一下如何制作好一张地图。

一、制图规范与风格

制图的设计要素要合理,对应布局也应尽量符合客观事实。之所以放在这一步讲制图规范与风格,主要是因为我们需要了解一下 VT 制图的机制才可以,毕竟制图并不是说做就做,部分操作的难度并不小。VT 的基础操作都已经贯通了,但真正的制图仍需不小的工程量,而且 VT 无法建模(因此自建网格需要用 3ds Max 来实现建模,此处不予介绍)。

做一张地图最重要的设计问题就是控制难度与长度。难度应该布局合理,并且有一定价值。技术难度可以根据制图意愿自主选择,但明显一些创新型技术点(如从机关之间飞越)就比只是把非对齐单轨/骑球等硬又无新意的技术要有更好的效果,创新型技术难点就算很难,也会吸引更多人去玩,因此对于制图中技术难度的把控应注重创新。至于逻辑难度,也应该注重逻辑的创新性和整体性,而不是通过频繁绕路来提高逻辑性,对于良好的逻辑性地图的评判就是如果看见一个对逻辑图通关视频,我们的第一印象是“哇,居然还能这么走”,而不是“看着 Ta 绕路我自己都觉得费劲”,如果逻辑设计的整体性大,自成体系的话,自然逻辑图的水平也会高很多,这就更考验制图者的设计力了。

对于地图的长度,应该把控合理,这方面可以参考一下原版关卡或一些好的制图作品,逐渐熟悉,就能良好把控,当然也可以根据主观意愿适当增长或缩短长度,但不应过度伸缩,导致效果欠佳。除此之外,地图符合客观事实也是很重要的,部分明显不合实际的设计应该否定。比如多层管状路面,如果加影子的话球在管内走影子也会投射到路面顶面,所以其实还不如不加影子好。总而言之,只要有一颗热衷于制图的心,把握好制图的态度并不断坚持,好地图就能出现。

二、制图的基本路线

对于新手乃至很多 VT 制图熟练的人,都推荐以下路线:

(1).导入模板关卡:使用任意关卡(建议原版第 10 关或 11 关,也可选择 180519 图等可利用度较高的自制地图,组和已有材质贴图较多)作为模板,开始自己的地图设计。

(2).删除多余元素：将模板地图中象征性的大型路块等多余物件及其对应网格删除，机关和独立风扇底座（原版关卡不存在）、云层和死亡区等物件以及模板地图中的材质和贴图等可以回收利用。（对于使用 3ds Max 自制网格的制图者，可以保留大路面/钢轨的物件，导入网格后直接替换网格，这样就省去了归组的步骤，如果是需要加影子的路面，还能这样直接加影子）。

(3).主体制图：导入自己设计规划中需要的素材，对素材进行拼接等综合制图操作，这部分也是制图的最主要部分（这点就需要考验对 VT 制图的整体感知能力和对地图的设计能力的，一些技巧还是很有用的）。归组、上材质乃至回收机关可能需要的小节组改换操作也需要在这一步完成。建议审查没有加影子的路面以加影子（强烈建议在构建地图之前就完成，否则将很麻烦）

(4).后期处理：不断测图和看情况改图，修复图中的问题（材质贴图问题以及影子问题如还未解决可以在这一步再作补充，但会更加麻烦），最后完工发布。

当然，也可以走如下路线：

(1).直接创建 cmo 文件或者利用模板（可在网络上下载，是一个只包含已有原版材质/贴图/组的 cmo 文件，还有一个用于加影子的无网格原版物件）

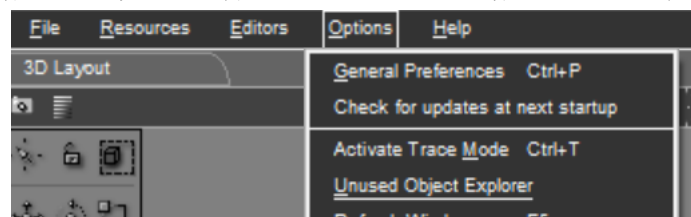
(2).导入素材，重复之前路线的(3)、(4)步骤：由于是直接创建新图，云层和死亡区等全部需要自己导入，而且需要导入更多素材来操作，还需要注意飞船只会出现在第 X 小节组（新建图里有 X 个小节组），测图时有可能也会感到疑惑。

因此用这种路线制图对新手有一定挑战性，但觉得可以尝试的人也可以试试，不过还是建议第一种制图路线。总之，有不忘初心的态度和制图的动力，即使因为制图水平不好而导致地图不是很完美，也会有含金量的。

附：地图的【栅（shān）格化】简介（概念提出者为 Nice_Melodies）：既然能通过设置物件的位置坐标来实现精确的移动，那么不妨让所有物体的坐标尽可能简单化，能是整数就尽量弄成整数，不能的也尽量弄成 2 的负整数倍（即小数部位类似 5,25,125 等的这些数）。这种方法已在 Ballance 制图的发展历程中被实践过，可以使地图做工更加精细，但需要花费的时间自然也更多。

三、关于地图占用存储的节约

制图还应节约其大小，风格可以不完全跟着原版走，但细节需做到尽可能好，精益求精。因此在制图测验结束后，首先应该删除不必要保留的物件，之后通过 Unused Object Explorer，将多余的网格、材质和贴图删除，



这是最基本的净化地图的方案。不过如果有多余不必要的光源和空组，需要将光源单独删除。

另外，很重要的一点是，地图后期处理时强烈建议把原版贴图都设成外置，只有可能用到的只能内置的非原版贴图才内置，这样原版贴图就可以从 **Textures** 根目录里读取，而不占用地图的大小。关于贴图存储格式的设置，已在之前的两节中提到。事实上，可以共用而无影响的内置贴图大量独立存在是导致地图占用存储非正常膨胀的主要原因。不过对于非机关类物件所需的网格，建议能不共用就不共用，虽然可做拼接制图的基础钢轨/路面的网格所占存储都占一定比重，但是我们之前就提到了路面共用网格会导致影子重复，钢轨共用网格会造成碰撞箱等不必要的麻烦，如果确定不再修改并且对地图实际上影响不大，可以本着减小地图大小的目的适当共用网格。当然，机关可以任意共用网格，如果最终确定不修改的话，还可以直接删掉机关的网格，不过建议慎用。

四、一些值得注意的制图细节补充

1.地图中对元素的命名虽然只有特殊几个元素有要求，但仍需注意。可以根据原版的命名格式来，一些不能遵从原版规律的元素也可以用比较好的仿照风格命名。对命名有要求的元素：云层物件的命名一定要是 **SkyLayer**，否则无法正常显示云层；显示出灯光光线的材质需命名为 **Laterne_Verlauf**，如果为其他命名则会导致灯柱的光线无法显示；需要归特殊物件组的物件的命名。

2.制图时绝不能让 **PC_Checkpoints** 组里没有物件，否则会导致玩图时火焰为三角形而且不能按 **esc** 退出，只能用任务管理器强行终结游戏程序。如需制作只有一小节的地图，可以正常归组第二小节的盘点火 **PC_TwoFlames_01**，然后将该盘点火放在地图中很远看不到的位置。小节组只需要保留一个就可以使飞船出现在第一节（有几个小节组，飞船就会出现在第几个小节组，所以需要制作几小节的地图，就需要保留几个小节组，当然需要引起注意的是正常游戏只能加载最多 8 小节的关卡，更多需要安装 **mod**，此处不再介绍）。

3.对于柱子的渐变和灯光的影子缺失的问题，原因很复杂，目前观点认为 **Ballance** 是 **VT2.1** 版本编辑的，而我们最低用的版本是 **3.0** 或 **3.5**，而找不到更以前的版本，而目前的版本会导致贴图像素格式出现通道失真问题。所以目前制图可以暂不考虑灯光的影子，对于柱子的渐变，可以替换通过之前的一个无意操作而导出的唯一“看起来渐变”的柱子素材来“实现”。（事实上，通过强制在地图中插入脚本来调整贴图像素格式从而获得原版渐变和灯影的方法已实现，步骤较为复杂，这里不再介绍）

4.归钢轨组的物件，其网格对应材质贴图会自动平滑，因此有非纯色平滑贴图又想表达钢轨质感的，归组时不能归钢轨组，但可以归路面组和钢轨声音组。

5.3 3D Entities 目录中的配置文件

在 Ballance/3D Entities 里有一些 nmo 配置文件,这些文件其实是伪装成 nmo 的 cmo 文件,而且 cmo 性质更加明显。这些文件决定着游戏的运行,让我们来简单分析一下其中的部分文件吧。如果要修改这部分文件,由于直接导出 nmo 无法保存内部的一些脚本和动画,为了避免损伤,一定要以 cmo 形式保存,最后改成 nmo。

此外,有些表格 (Arrays) 设置了 IC,因此编辑完后要右键重新设置一下 IC (Initial conditions), 不然编辑的内容无法正常保存。

marker: 里面是文件夹的名字,具体作用未知。

/Level: 关卡文件 (Level_xx.nmo)。

/PH: 模型的原型文件。

AnimTrafo.nmo 变球器本体和动作描述文件。

Balls.nmo 球的物理信息描述文件,以及球的阴影,碎片等效果。

Camera.nmo 摄像机以及游戏中分数和剩余球数显示的界面。

Gameplay.nmo 游戏的最核心脚本。负责处理游戏过程中的全部信息。

Intro.nmo 游戏的开头画面。

Language.nmo 语言描述文件,包含游戏内的语言字符串和键盘键名。

Levelinit.nmo 关卡初始化文件,包含搭建关卡环境的脚本,路面的物理信息和关卡信息等。

Menu.nmo 菜单描述文件,包含所有的菜单 2D 帧以及脚本。

MenuLevel.nmo 著名的菜单关卡。无数外挂都是从这里开放脚本注入的。

Musicfiles.nmo 音乐文件的定义。

Sound.nmo 游戏声音以及背景音乐的脚本。

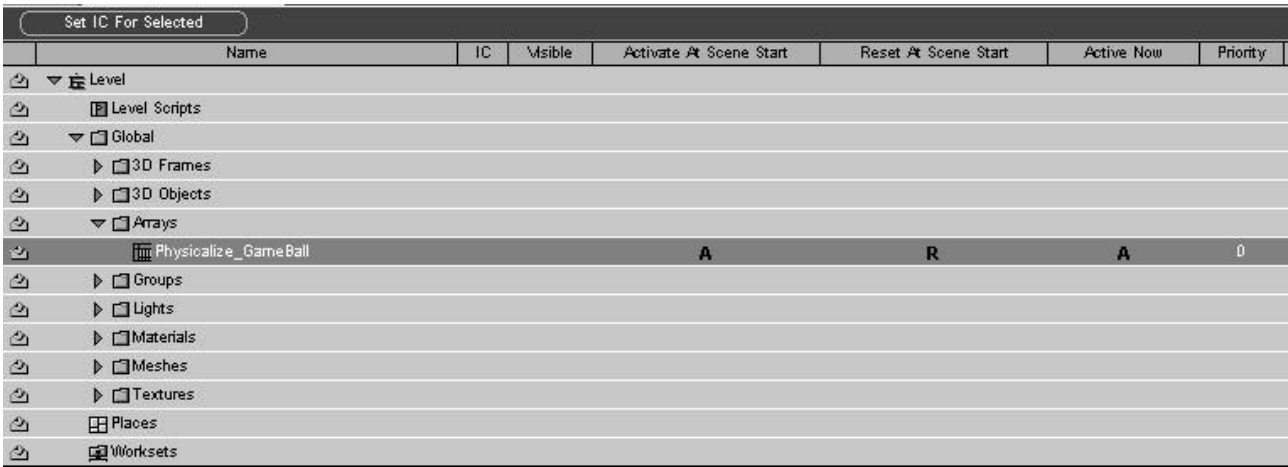
Tutorial.nmo 描述第一关教程的 3D 帧和图形。

其实还有一个文件,叫做 base.cmo (该文件其实是个 vmo 文件,无法直接用 VT 打开,不过目前已经有大神将此文件破解),在游戏根目录下,它是播放器打开时加载的第一个文件,负责加载其他的基本组件。

如下是部分文件的较详细的介绍:

1.Balls.nmo

我们打开它，会发现上面有一个 3D 视图，但目前尚未知道改动的结果。所以改动球的设定，主要是改动 Level Manager 里面的应该表格数据，该数据的位



置如下图所示，双击可以看见一张展示球性质的表格。

如图所示，通过改动表格的内容，就可以改变球的性质了，如下是几个容易理解的可改动因素：

	0 : Ballname	1 : Friction	2 : Elasticity	3 : Mass	4 : CollGroup	5 : Linear Damp	6 : Rot Damp	7 : Force
0	Ball_Paper	0.5000	0.4000	0.2000	Ball	1.5000	0.1000	0.0650
1	Ball_Stone	0.5000	0.1000	10.0000	Ball	0.3000	0.1000	0.9200
2	Ball_Wood	0.8000	0.2000	1.9000	Ball	0.9000	0.1000	0.4300

Friction: 摩擦力 Elasticity: 弹力 Mass: 质量

Coll Group: 碰撞组，相同碰撞组的物体会互相无视

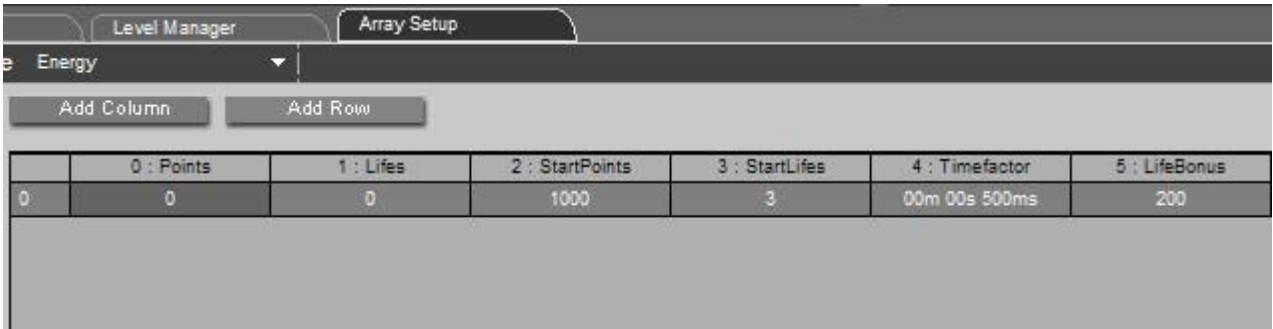
Linear Damp（Linear Speed Dampening）：移动阻力

Rot Damp（Rotate Speed Dampening）：旋转阻力

Force: 按方向键给予球的移动力量（推力），这里是改球速所在的具体位置。比如要把全部球都改成 2 倍速，就自上而下填入：0.1300、1.8400、0.8600

2.GamePlay.nmo，里面能改的也只有 Array（表格）了。

总共 5 个表格，其中改了有用的应该只有 Energy 这个了。



Points 和 Lives: 当前的分数和生命值, 可以无视

StartPoints: 游戏起始分数 StartLives: 起始生命

Timefactor: 每过多长时间扣一分 m=分 s=秒 ms=毫秒

LifeBonus: 每一个生命加多少分

3.Camera.nmo

如果把里面的 2D Entities 删掉就能不在游戏中显示生命/分数

4.Levelinit.nmo

这里的表格多, 但多数未被玩家开发利用。其中两个比较有用, 这两个分别是 AllLevel 和 Physicalize_Floors。

AllLevel 组决定了: 出生的球 StartBall、天空 Sky、亮度 Light、关卡奖励分 LevelBonus、音乐 Music

而 Physicalize_Floors 组决定了路面 Phys_Floors

和钢轨 Phys_FloorRails 的摩擦、弹力等 (这个组可以新增自定义类型的路面, 并且自主设置其摩擦、弹力等, 比如新增弹力路面后把得到的 Levelinit.nmo 做成弹力路面补丁, 就可以在编辑的地图中设置弹力路面的组了)。

5.Language.nmo, 这个文件决定了 ballance 各种语言所显示的内容。在 language 组里, 第 0 列为要表达的内容, 1~5 列依次为德语、英语、西班牙语、意大利语和法语。

后记

编著本教程的想法于 2018 年 6 月诞生，目的是更一体化地介绍使用 Virtools 来制作和修改 Ballance 地图，并且提供少量改变游戏设置的方法。最初的代表性版本在 2018 年 7 月发行，书名为《TCBV》（《Textbook of Changing Ballance by Virtools》），后来经过不断改善，与 2018 年 9 月 30 日发行新版本《VT 制图基础 2A》，最后于 2018 年 11 月 30 日合并和整理了一些新的内容，自此《Virtools 制图》第一版正式发行。此版本是于 2019 年 2 月 11 日再次修改的新版本，尽量做到更加贴近制图。

第一版封面介绍：

本书旨在综合运用 Virtools 修改和制作 Ballance 地图。通过认真理解本书对于 Virtools 制图的整体观和具体操作，读者可以领略 Virtools 制图，并且有能力做出 Ballance 地图。

第一版参考文献：

1. 《Ballance 修改手册》，作者 jxpxxzj，2009 年 3 月
2. 《virtools 制图手册》，作者 Chris Xiong，2014 年 8 月
3. 《Virtools 使用教程》，作者平衡总统，2016 年 2 月
4. 《Ballance 修改教程（一）纯 VT 制图》，作者 Nice Melodies，2016 年 7 月

给此教程做出过贡献的人：

GEcoldstar、BallanceBug、Chris Xiong、Nice Melodies、jxpxxzj、2jjy、zzq_203、梨隼 washing、Sterd、平衡总统